GOODAT: Towards Test-Time Graph Out-of-Distribution Detection

Luzhi Wang[1], Dongxiao He[1], He Zhang[2], Yixin Liu[2], Wenjie Wang[3], Shirui Pan[4*], Di Jin[1], Tat-Seng Chua[3]

[1]College of Intelligence and Computing, Tianjin University
[2]Faculty of Information Technology, Monash University
[3]School of Computing, National University of Singapore
[4]School of Information and Communication Technology, Griffith University
{wangluzhi, jindi, hedongxiao}@tju.edu.cn, {he.zhang1, yixin.liu}@monash.edu,
wenjiewang96@gmail.com, s.pan@griffith.edu.au, dcscts@nus.edu.sg

Abstract

Graph neural networks (GNNs) have found widespread application in modeling graph data across diverse domains. While GNNs excel in scenarios where the testing data shares the distribution of their training counterparts (in distribution, ID), they often exhibit incorrect predictions when confronted with samples from an unfamiliar distribution (out-of-distribution, OOD). To identify and reject OOD samples with GNNs, recent studies have explored graph OOD detection, often focusing on training a specific model or modifying the data on top of a well-trained GNN. Despite their effectiveness, these methods come with heavy training resources and costs, as they need to optimize the GNN-based models on training data. Moreover, their reliance on modifying the original GNNs and accessing training data further restricts their universality. To this end, this paper introduces a method to detect **G**raph **O**ut-**o**f-**D**istribution **A**t **T**est-time (namely GOODAT), a *data-centric*, *unsupervised*, and *plug-and-play* solution that operates independently of training data and modifications of GNN architecture. With a lightweight graph masker, GOODAT can learn informative subgraphs from test samples, enabling the capture of distinct graph patterns between OOD and ID samples. To optimize the graph masker, we meticulously design three unsupervised objective functions based on the graph information bottleneck principle, motivating the masker to capture compact yet informative subgraphs for OOD detection. Comprehensive evaluations confirm that our GOODAT method outperforms state-of-the-art benchmarks across a variety of real-world datasets.

Introduction

Graph neural networks (GNNs) are potent representation learning methods that focus on processing graph data (Wang et al. 2019a,b), and have been widely used in financial networks (Zheng et al. 2021), binary code analysis (Wang et al. 2023a), sarcasm detection (Wang et al. 2023b; Yu et al. 2023), etc. Generally, GNNs provide strong support for accurate prediction of downstream tasks by capturing the distribution of training data (Kipf and Welling 2017; Zhang et al. 2021). However, when these well-trained GNN models are deployed in open-world scenarios, they inevitably encounter graph samples from unknown classes, the so-called graph "Out-of-Distribution (OOD)" data (Liu et al. 2023a). On the OOD data, well-trained GNNs may not be effective, as the features and distribution patterns of these OOD graphs are not exposed during GNN training (Bai et al. 2023). This situation can lead to prediction errors when dealing with these unknown distribution samples, thereby reducing the reliability of GNNs. In this scenario, an ideal GNN model should possess the capability to identify and reject OOD samples, rather than misclassifying them as belonging to the in-distribution (ID) classes.

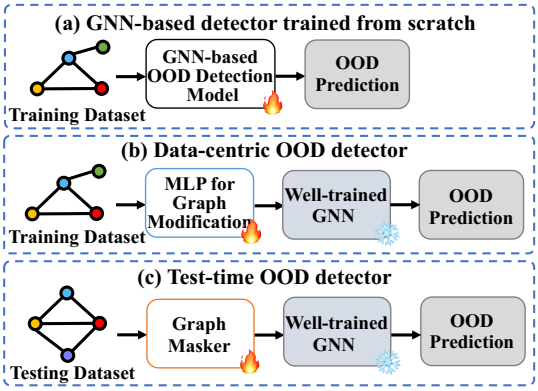


Figure 1: Comparisons between GOODAT and other methods. To detect OOD samples, **(a)** most GNN-based methods need to learn a detector from the training data (Liu et al. 2023a); **(b)** other data-centric methods learn an MLP to modify the training data while keeping the well-trained GNN fixed (Guo et al. 2023). **(c)** In contrast, our test-time OOD detector directly works on the test data without needing to consult the training data and change the parameters of the well-trained GNN.

To effectively identify OOD graph samples, various graph OOD detection approaches have emerged (Huang, Wang, and Fang 2022; Hoffmann, Galke, and Scherp 2023). A subset of research (Liu et al. 2023a) centers around crafting GNN-based OOD detection models that are meticulously tailored for graph OOD detection tasks. While demonstrating effectiveness, they need to train additional GNNs from scratch, resulting in a heavy resource expenditure. Another

*Corresponding author.

type of methodology solves the graph OOD detection problem from a data-centric perspective (Zheng et al. 2023a), implementing downstream tasks by modifying data on top of well-trained GNNs. As a typical method, AAGOD (Guo et al. 2023) imposes a multi-layer perception (MLP) based parametric matrix on the adjacency matrix of each training graph. This is done without changing the parameters of the well-trained GNN, effectively widening the differences between OOD and ID graphs. By merely optimizing the MLP instead of retraining the GNN, AAGOD mitigates the effort required for model design and parameter training.

Despite the effectiveness of the aforementioned methods, both types of existing graph OOD detection approaches rely on the training dataset, leading to several limitations. Firstly, training a task-specific GNN-based model for OOD detection usually requires significant *computational resources and costs,* e.g., training an additional GNN from scratch (Jin et al. 2023c; Gui et al. 2022). Secondly, in certain scenarios or platforms where model architectures are inaccessible due to privacy concerns (Zheng et al. 2023c), making modifications or adjustments to the GNN architecture becomes impractical. Meanwhile, in some cases (e.g., federated learning (Tan et al. 2023a)), the original training dataset may also be inaccessible (Tan et al. 2023b), further obstructing the training process of these OOD methods. In such instances, these aforementioned OOD detection methods cannot be applied, exposing their *limited universality*.

To address the above issues, we delve into the research problem of **test-time graph OOD detection**. Concretely, a test-time graph OOD detection method solely learns on testing data, without being dependent on training data. Moreover, the method is not expected to redesign the backbone model or add additional networks, ensuring its adaptability across diverse well-trained GNN models, irrespective of the characteristics of the training data. Nevertheless, developing such a test-time OOD detection method presents substantial complexities due to the following challenges. *Challenge 1: Inconsistent learning objective.* Most well-trained GNNs are trained for specific graph learning tasks (e.g., graph classification) rather than OOD detection. In this case, how to align these pre-existing models with the target of OOD detection remains a difficulty. *Challenge 2: Absence of labels.* During test time, the lack of graph labels poses a challenge in unsupervised detecting ID and OOD graphs, compelling the need to design an unsupervised model. *Challenge 3: Unavailability of training data.* Constrained by the test-time setting, access to the training dataset is unfeasible. This scarcity of comprehensive knowledge about the original training data of the GNN model imposes significant limitations on our capacity to integrate an OOD detection model into GNNs.

To solve the aforementioned challenges, we propose a novel data-centric method to detect **G**raph **O**ut-**O**f-**D**istribution **A**t **T**est-time, namely GOODAT. To address *challenge 1*, we first construct a plug-and-play graph masker consisting of parameterized matrices. This enables us to compress informative subgraphs from the original input graphs, thereby indicating their ID or OOD nature. This lightweight masker can seamlessly integrate with any well-trained GNN, endowing it with the capability to detect OOD samples. To handle *challenge 2*, we design three unsupervised loss functions based on the graph information bottleneck (GIB) principle, guiding the masker to capture compact yet sufficient subgraphs for distinguishing ID and OOD graphs. To deal with *challenge 3*, we fully exploit the test data to capture the ID graph patterns of training data. Specifically, we operate under the assumption that all graphs in the test dataset are inherently ID. Guided by these surrogate ID labels, the OOD subgraph compressed by the ID label should significantly differ from the ID subgraph compressed by the ID label. This distinction serves as a reliable basis for effectively detecting OOD graphs. Fig. 1 shows the differences between GOODAT and other methods. To sum up, the main contributions of this paper are three-fold:

- **New paradigm.** We pioneer the learning paradigm of test-time graph OOD detection, unveiling a fresh perspective. This innovative paradigm sheds light on *lightweight*, *training data-independent*, and *plug-and-play* solutions for graph OOD detection, seamlessly applicable to any well-trained GNN models.
- **Novel method.** We propose a simple yet effective method, GOODAT, to solve the test-time graph OOD detection problem. Leveraging the information bottleneck principle, GOODAT captures informative subgraphs from each input graph, thus enabling the accurate identification of OOD samples within the test dataset.
- **Extensive experiments.** We conduct experiments on multiple datasets and scenarios to verify the effectiveness and superiority of GOODAT. Experimental results show that GOODAT has achieved significant improvements in graph OOD detection tasks compared to baselines.

Related Works

Graph Neural Networks

With graph-related problems rising in various real-world scenarios (Wu et al. 2022; Zheng et al. 2023b; Wu et al. 2023a), GNNs have emerged as a powerful paradigm for tackling complex graph data (Liu et al. 2023c; Zheng et al. 2023d, 2022). GNNs have shown remarkable success across various domains (Zhang et al. 2022), including social networks (Zheng et al. 2022), anomaly detection (Liu et al. 2023b), binary code analysis (Jin et al. 2022), and recommender systems (Jin et al. 2023b,a). Although many methods have been proposed to improve GNN performance (Kipf and Welling 2017; Xu et al. 2019), concerns have emerged about other aspects (e.g., robustness (Zhang et al. 2023c), privacy (Zhang et al. 2023a; Wu et al. 2024), fairness (Zhang et al. 2023b)) of GNN models. In the context of generalization, while GNNs perform well on ID data, they may perform poorly on OOD data (Liu et al. 2023a).

Graph OOD Detection

Graph OOD detection aims to detect whether the test graph is ID or OOD. Recently, many studies have been proposed (Zhao et al. 2020; Stadler et al. 2021; Wu et al. 2023b). For example, GOOD-D (Liu et al. 2023a) uses graph contrastive

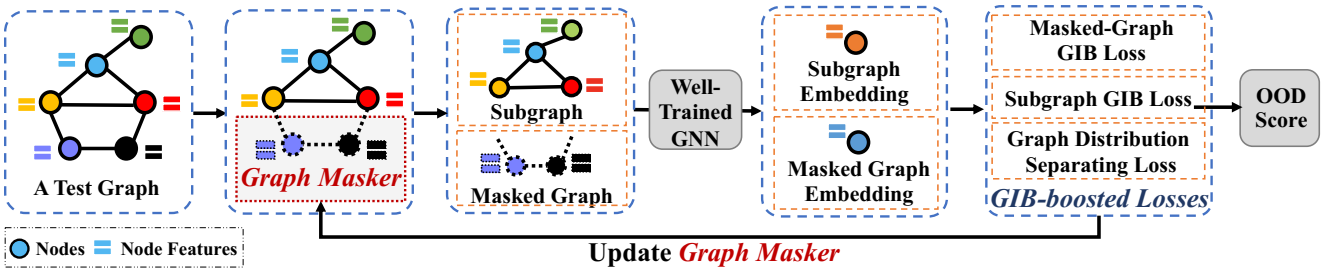


Figure 2: Overview of GOODAT. In the GOODAT training process, a graph masker M is applied on the input test graph G, consisting of two parameterized matrices. This graph masker M is trained by utilizing three GIB-boosted losses, taking the graph G and its corresponding surrogate ID label Y as inputs. The informative subgraph Z and the masked graph Z' are obtained with the trainable parameters M (e.g., $Z = G \odot M$). During the inference phase of target GNNs, the OOD score of a test graph is obtained by the graph masker and GIB-boosted losses to infer if the input graph is an OOD graph.

learning to provide an unsupervised view of graph OOD detection. GraphDE (Li et al. 2022) models the graph generative process to learn latent environment variables for detection. OODGAT (Song and Wang 2022) utilizes a multi-head attention mechanism to compute node weights and transform them into edge weights, aiding in the identification of OOD nodes. AAGOD (Guo et al. 2023) introduces a data-centric framework that enlarges the differences between OOD and ID graphs. However, the above methods rely on the training dataset to train OOD detection models, which is different from our GOODAT method.

Preliminaries and Problem Definition

Graphs and **Graph Maskers.** Given an undirected graph $G = (X, A)$, $X \in \mathbb{R}^{n \times d}$ represents the node feature matrix with n nodes and each node has a feature dimension of d, and $A \in \mathbb{R}^{n \times n}$ indicates the adjacency matrix of G. A label Y is associated with G, where $Y = 0$ indicates that the graph is ID, and $Y = 1$ indicates that the graph is OOD. A graph masker is defined as $M = (M_X, M_A)$, where $M_X \in \mathbb{R}^{n \times d}$ and $M_A \in \mathbb{R}^{n \times n}$ are parameterized matrices for extracting the subgraph from the original graph G. We can modify the graph masker by gradient descent on M_X and M_A. For example, we can obtain a subgraph $Z = G \odot M = (X \odot M_X, A \odot M_A)$ from G, where $\odot$ denotes the Hadamard product. Given Z, the remaining part of the test graph is donated as the masked graph $Z' = (X - X \odot M_X, A - A \odot M_A)$. For a given test graph $G = Z \cup Z'$, the overlap between Z and Z' is defined as $Z \cap Z'$, and $|Z \cap Z'|$ represents its size.

Graph Information Bottleneck (GIB). From the view of information theory, the information bottleneck aims at compressing the original information to obtain crucial information related to the label (Alemi et al. 2017). As for graph information bottleneck, given a graph G and its label Y, the graph information bottleneck aims to compress the information of G to obtain a compressed graph Z, which maximizes the mutual information between Y and Z and minimizes the mutual information between Z and G (Yu et al. 2021). Specifically, assuming that $I(\cdot)$ indicates the Shannon mutual information, the GIB can be defined as (Wu et al. 2020):

$$\max_{Z} I(Y, Z) - \lambda I(G, Z), \tag{1}$$

where λ is a Lagrange multiplier.

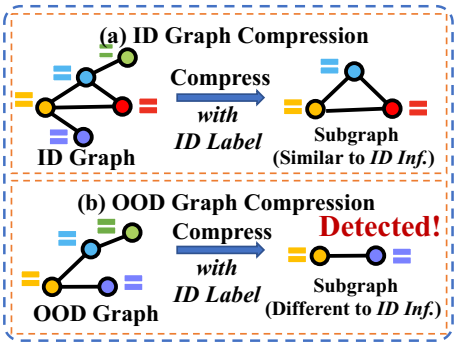


Figure 3: GOODAT intuition. "Inf." denotes information.

Test-time Graph OOD Detection. For a test graph G, we assumed that it comes from the ID or OOD graph distribution. The test-time graph OOD detection task is defined as:

Definition 1 (Test-time graph OOD detection)**.** *Given a well-trained GNN f and a graph G from the test dataset, the test-time graph OOD detection aims to determine the source distribution of G during the inference time of f with an OOD detector $\mathtt{D}$. Specifically, the objective of the detection task is:*

$$\mathtt{Detection\ label} = \begin{cases} 1\ (OOD), & if\ \mathtt{D}(f, G) \geqslant \eta \\ 0\ (ID), & if\ \mathtt{D}(f, G) < \eta \end{cases} \tag{2}$$

where η is a threshold, and the parameters of f are fixed during the OOD detection.

Methods

Fig. 2 shows the overview of our GOODAT method, whose effectiveness in distinguishing ID and OOD graphs is demonstrated below. Given the surrogate label, GIB is employed to find the subgraph with the highest correlation to its label for each input graph. Hence, the subgraphs obtained

from the ID and OOD graphs can be distinguishable, since they come from different distributions. During the test-time, a well-trained GNN f tends to make right predictions on ID graphs while making wrong predictions on OOD graphs, since OOD graphs can be regarded as being in an unknown class. In this case, the subgraph of OOD graphs extracted by the GIB principle can be significantly different from the ones of ID graphs. As shown in Fig. 3, considering an ID graph G^I with the right label prediction (e.g., $Y = 0$), the subgraph obtained from GIB is denoted as Z_0^I. When the predicted label (e.g., $Y = 0$) is wrong (e.g., for an OOD graph G^O), the extracted subgraph Z_0^O obviously differentiates from Z_0^I as Z_0^I is not included in G^O. Otherwise, G^O is not an OOD graph. Such distinguishability between Z_0^I and Z_0^O can effectively separate ID and OOD graphs, enabling our method to execute the test-time OOD detection.

In this paper, three different GIB-boosted loss functions are presented to enhance the capability of GIB on reasoning subgraphs that most correlate with labels. Below are details on each of the three loss functions.

Subgraph GIB Loss

From the perspective of factual reasoning, the subgraph GIB loss facilitates the extraction of informative subgraph Z that related to the predicted label. With the surrogate label Y and obtaining the informative subgraph $Z = G \odot M$ by applying the graph masker $M = (M_X, M_A)$, we propose to utilize the following subgraph GIB loss to optimize the parameters M_X and M_A in M. Specifically, we maximize the Shannon mutual information between the subgraph Z (embedding from f) and the label Y, while minimizing the Shannon mutual information between the subgraph Z and the original test graph G (embedding from f). We seek the most informative and compressed subgraph representation by optimizing the following objective:

$$\max_{Z} I(Z, Y) - \alpha I(Z, G), \tag{3}$$

where α is the Lagrange multiplier to balance the two components. Building on the work (Alemi et al. 2017), we transfer $I(Z, Y)$ and $I(Z, G)$ into our loss function.

(1) According to the definition of mutual information, we obtain $I(Z, Y) = \iint p(Z, Y) \log \frac{p(Y|Z)}{p(Y)} dY dZ$. For the item $p(Y|Z) = \int \frac{p(Y|G)p(Z|G)p(G)}{p(Z)} dG$, which depends on the Markov property (Sun et al. 2022) and is difficult to calculate, we propose using $q(Y|Z)$ to approximate $p(Y|Z)$. Specifically, we use the Kullback-Leibler divergence (Kullback and Leibler 1951) to measure their distance and make them closer. Since $D_{KL}(p(Y|Z)||q(Y|Z)) \geqslant 0$, we obtain

$$\int p(Y|Z) \log p(Y|Z) dy \geqslant \int p(Y|Z) \log q(Y|Z) dY. \tag{4}$$

According to Eq. (4), the lower bound of $I(Z, Y)$ is:

$$\begin{aligned} I(Z, Y) &\geqslant \iint p(Y, Z) \log \frac{q(Y, Z)}{q(Z)} dZ dY - \int p(Y) \log p(Y) dY. \\ &\geqslant \iiint \frac{p(Y, G)p(Z, G)}{p(G)} \log \frac{q(Y, Z)}{q(Z)} dZ dY dG. \end{aligned} \tag{5}$$

(2) For $I(Z, G) = \iint p(Z, G) \log \frac{p(Z,G)}{p(Z)p(G)} dZ dG$, where it is not easy to directly calculate $p(Z)$, we propose to use a learnable $\phi(Z)$ to approximate $p(Z)$. Similarly, since $D_{KL}(p(Z)||\phi(Z)) \geqslant 0$, an upper bound of $I(Z, G)$ is:

$$I(Z, G) \leqslant \iint p(Z, G) \log \frac{p(Z, G)}{\phi(Z)p(G)} dG dZ. \tag{6}$$

Combining the inequalities about $I(Z, Y)$ and $I(Z, G)$, we derive a lower bound of $I(Z, Y) - \alpha I(Z, G)$. Specifically,

$$\begin{aligned} &I(Z, Y) - \alpha I(Z, G) \\ &\geqslant \iiint \frac{p(Y, G)p(Z, G)}{p(G)} \log \frac{q(Y, Z)}{q(Z)} dZ dY dG \\ &- \alpha \iint p(Z, G) \log \frac{p(Z, G)}{\phi(Z)p(G)} dG dZ. \end{aligned} \tag{7}$$

where the hyperparameter α balances informativeness and compression. Instead of directly solving $\max_Z I(Z, Y) - \alpha I(Z, G)$, we employ the following subgraph GIB loss:

$$\begin{aligned} \mathcal{L}_s =& \frac{1}{N} \sum_{i=1}^{N} (\mathbb{E}_{\xi \sim p(\xi)}(-\log q(Y_i|Z_i)) + \alpha D_{KL}[p(Z_i|G_i)||\phi(Z_i)]), \\ &\approx \mathcal{L}_{cls}(q(Y_i|Z_i), Y_i) + \alpha D_{KL}[p(Z_i|G_i)||\phi(Z_i)]), \end{aligned} \tag{8}$$

where N represents the number of graphs, $\mathcal{L}_{cls}$ indicates the classification loss. We use the Gaussian distribution to approximate q and ϕ. Fig. 4 (a) illustrates the impact of subgraph GIB loss on the subgraph compression process.

Masked Graph GIB Loss

From the perspective of counterfactual reasoning, the masked graph GIB loss is used to obtain the masked graph Z', which ensures the irrelevance between the Z' and the label. Considering that Z' is regarded as the noise part of the graph G, the optimization objective from the GIB perspective can be formulated as:

$$\max_{Z'} \beta I(Z', G) - I(Z', Y), \tag{9}$$

where β is a Lagrange multiplier. Similar to the derivation of the subgraph GIB loss, the masked graph GIB loss $\mathcal{L}_m$ can be defined as:

$$\begin{aligned} \mathcal{L}_m =& \frac{1}{N} \sum_{i=1}^{N} (\mathbb{E}_{\xi \sim p(\xi)}(\log q(Y_i|Z_i') - \beta D_{KL}[p(Z_i'|G_i)||\phi(Z_i')]), \\ &\approx -\mathcal{L}_{cls}(q(Y_i|Z_i'), Y_i) - \beta D_{KL}[p(Z_i'|G_i)||\phi(Z_i')]). \end{aligned} \tag{10}$$

We use the Gaussian distribution to approximate q and ϕ. Fig. 4 (b) illustrates the masked graph GIB loss.

According to an existing study (Tan et al. 2022), only considering the subgraph GIB loss $\mathcal{L}_s$ leads to sufficient but not necessary Z, while exclusively using the masked graph GIB loss $\mathcal{L}_m$ tends to generate necessary but not sufficient Z. To overcome this limitation (i.e., a large $|Z \cap Z'|$), employing $\mathcal{L}_s$ and $\mathcal{L}_m$ simultaneously helps to dig out sufficient and necessary Z (i.e., limited $|Z \cap Z'|$).

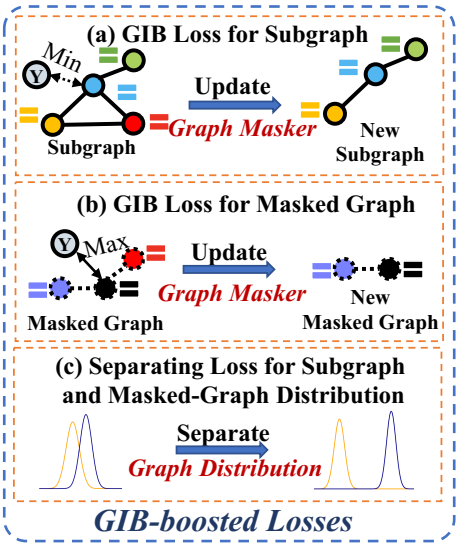


Figure 4: Illustrations of the three losses in GOODAT.

Graph Distribution Separating Loss

To further reduce the overlapping size (i.e., $|Z \cap Z'|$), we propose the graph distribution separating loss to separate the distributions of Z and Z' from the statistical view, where Z and Z' are supposed to own a small joint probability density. Sklar (1959) declares that the joint distribution of N random variables can be decomposed into the respective marginal distributions of the N variables and a Copula function (Sklar 1973), so as to separate the randomness and coupling of the variables. In this paper, we assume that Z and Z' have the following Gaussian form of marginal distributions, i.e.,

$$p_Z(Z) = \frac{1}{\sqrt{2\pi}|\sigma_1|} e^{\frac{-Z^2}{2\sigma_1^2}}, p_{Z'}(Z') = \frac{1}{\sqrt{2\pi}|\sigma_2|} e^{\frac{-Z'^2}{2\sigma_2^2}}, \quad (11)$$

where σ_1/σ_2 represents the standard deviation of Z/Z'.

According to the Copula theory (Durante, Fernandez-Sanchez, and Sempi 2013), there must be a function that couples the marginal probabilities. Therefore, the joint probability function of Z and Z' can be defined as:

$$\mathcal{L}_d = \frac{1}{2\pi|\sigma_1\sigma_2|\sqrt{1-\rho^2}} e^{\left[-\frac{1}{2(1-\rho^2)} \times \left[\frac{Z^2}{\sigma_1^2} - 2\rho\frac{Z \times Z'}{|\sigma_1\sigma_2|} + \frac{Z'^2}{\sigma_2^2}\right]\right]}, \quad (12)$$

where ρ indicates the correlation coefficient. Fig. 4 (c) illustrates how the graph distribution separating loss influences the distinguishability of graph distributions.

Training of GOODAT and OOD Graph Detection

By integrating the above three loss functions together, the overall loss $\mathcal{L}_g$ for training M can be expressed as:

$$\mathcal{L}_g = \mathcal{L}_s + \mathcal{L}_m + \mathcal{L}_d. \quad (13)$$

Note that in our method, all parameters in the target GNN f are fixed, and we employ the surrogate label Y and embeddings of G, Z, and Z' to train the mask generator M.

When implementing OOD detection, we employ the subgraph GIB loss (Eq. 8) as the graph OOD detection score. Specifically, for a test graph G, we first use the well-trained graph masker M to obtain its informative subgraph $Z = G \odot M$. Next, we obtain the GIB loss on Z w.r.t. $\mathcal{L}_s$, which is used to determine whether G is an OOD graph by Eq. (2).

Experiments

Experimental Setup

Datasets In this paper, we utilize the evaluation protocol proposed by (Liu et al. 2023a) which encompasses a graph OOD detection benchmark and a graph anomaly detection benchmark. The graph OOD detection benchmark contains 8 pairs of molecule datasets, 1 pair of bioinformatics datasets, and 1 pair of social network datasets. Each dataset pair within the same field exhibits a moderate domain shift. Following the setting in previous studies (Liu et al. 2023a; Guo et al. 2023), we allocate 90% of ID samples for training GNNs, while the remaining 10% of ID samples and an equal number of OOD samples constitute the test set. The graph anomaly detection benchmark comprises 15 datasets from TU benchmark (Morris et al. 2020). Anomalies encompass samples of the minority or real anomalous class, with the rest categorized as normal data. In our test-time OOD detection setting, we use the test set of each dataset as the input of GOODAT for test-time training.

Baselines & Settings We compared GOODAT with 13 mainstream baseline methods for graph OOD detection, which can be divided into four categories.
(1) Graph Kernels + Detectors methods use graph kernels as the pre-train model to learn graph embedding, and then inputs these graph embeddings to OOD/anomaly detector. Graph kernels include the Weisfeiler-Lehman kernel (WL) (Shervashidze et al. 2011) and propagation kernel (PK) (Neumann et al. 2016). OOD/anomaly detectors include local outlier factor (LOF) (Breunig et al. 2000), one-class SVM (OCSVM) (Manevitz and Yousef 2001), and isolation forest (iF) (Liu, Ting, and Zhou 2008).
(2) Graph Neural Networks + Detectors methods utilize self-supervised GNNs as pre-trained models with OOD detectors for detection tasks. The self-supervised GNN models include InfoGraph (Sun et al. 2020) and GraphCL (You et al. 2020). The detectors include iF and Mahalanobis distance (MD) (Sehwag, Chiang, and Mittal 2021).
(3) Test-time Training Methods achieve graph OOD generalization at test-time, without modifying the parameters of well-trained GNNs. The most recent method that falls into this category is GTrans (Jin et al. 2023c). Since GTrans is not explicitly designed for graph OOD detection, its loss value is adopted as the OOD score in our experiments.
(4) Data-centric Methods. One quintessential method of this category is AAGOD (Guo et al. 2023). AAGOD employs a graph adaptive amplifier module, which is integrated into a well-trained GNN to facilitate graph OOD detection. Unlike the test-time training approach, AAGOD employs the training set for the OOD detection training process. As of the submission of this paper, AAGOD has not released the code, so we conduct comparisons based on the experimental

ID dataset	ENZYMES	Tox21	FreeSolv	BBBP	ClinTox	Esol	*Avg.* *
OOD dataset	PROTEIN	SIDER	ToxCast	BACE	LIPO	MUV	*Rank*
PK-LOF	50.47±2.87	51.33±1.81	49.16±3.70	53.10±2.07	50.00±2.17	50.82±1.48	9.9
PK-OCSVM	50.46±2.78	51.33±1.81	48.82±3.29	53.05±2.10	50.06±2.19	51.00±1.33	10.0
PK-iF	51.67±2.69	49.87±0.82	52.28±1.87	51.47±1.33	50.81±1.10	50.85±3.51	8.1
WL-LOF	52.66±2.47	51.92±1.58	51.47±4.23	52.80±1.91	51.29±3.40	51.26±1.31	7.2
WL-OCSVM	51.77±2.21	51.08±1.46	50.38±3.81	52.85±2.00	50.77±3.69	50.97±1.65	8.1
WL-iF	51.17±2.01	50.25±0.96	52.60±2.38	50.78±0.75	50.41±2.17	50.61±1.96	9.3
InfoGraph-iF	60.00±1.83	56.28±0.81	56.92±1.69	53.68±2.90	48.51±1.87	54.16±5.14	5.4
InfoGraph-MD	55.25±3.51	59.97±2.06	58.05±5.46	70.49±4.63	48.12±5.72	77.57±1.69	4.5
GraphCL-iF	61.33±2.27	56.81±0.97	55.55±2.71	59.41±3.58	47.84±0.92	62.12±4.01	5.7
GraphCL-MD	52.87±6.11	58.30±1.52	60.31±5.24	75.72±1.54	51.58±3.64	78.73±1.40	2.6
GTrans	49.94±5.67	61.67±0.73	50.81±3.03	64.02±2.10	58.54±2.38	76.31±3.85	5.5
AAGOD-GIN$_S$+	66.22	64.26	–	67.80	–	–	–
AAGOD-GIN$_L$+	65.89	57.59	–	57.13	–	–	–
Ours	**66.29±1.54**	**68.92±0.01**	**68.83±0.02**	**77.07±0.03**	**62.46±0.54**	**85.91±0.27**	**1.4**

Table 1: OOD detection results in terms of AUC score (%). *Full results are available at arXiv:2401.06176 due to the page limitation.

results outlined in the AAGOD evaluations. AAGOD exists in two versions: AAGOD-GIN$_S$+ and AAGOD-GIN$_L$+, which correspond to distinct OOD evaluation methods. Unreported experimental results are denoted by '–'.

We employ a GIN (Xu et al. 2019) as the well-trained GNN encoder. We use the Adam optimizer (Kingma and Ba 2014) for optimization. Experiments run on a GeForce GTX TITAN X GPU with 24 GB memory, repeated five times for average scores and standard deviations. The best results are highlighted with **bold**. Experimental analyses are based on the **full** results.

Performance of Graph OOD Detection

We conduct a comparative analysis of our proposed method against 13 competing approaches across 10 graph OOD detection datasets. The results, in terms of AUC scores, are summarized and presented in Table 1. From this comprehensive comparison, we garner the following insights: 1) GOODAT showcases a remarkable performance by outperforming all baseline methods on 8 datasets. Furthermore, when considering the average rank across all methods, our proposed approach stands as the leader. This underscores the effectiveness of GOODAT in accurately detecting OOD samples within diverse graph-structured datasets. 2) While we may not have achieved the absolute best results on two datasets, our performance is in close proximity to the optimum results. One plausible explanation for this outcome is the relatively high edge density observed in these datasets, which potentially reduces the effect of GIB-boosted loss for subgraph compression. 3) Although the test-time graph OOD generalization method GTrans is also effective for OOD detection, our method has achieved overall advantages on all datasets. This demonstrates that directly substituting of graph OOD generalization for graph OOD detection may lead to sub-optimal performance. 4) Compared to the AAGOD method, GOODAT shows superiority across all datasets. This implies that our method achieves better results only using a test dataset, showcasing enhanced efficacy.

Performance of Graph Anomaly Detection

To assess the potential applicability of GOODAT on graph anomaly detection tasks, we conduct experiments on 15 datasets, following the evaluation protocol in (Liu et al. 2023a; Ma et al. 2022). We select 6 graph anomaly detection methods and a test-time training method as baselines, and the experiment results are summarized in Table 2. The AAGOD baseline is not included here as its code is not released. Experimental results indicate that GOODAT's applicability extends seamlessly to the anomaly detection scenario, showcasing remarkable performance. This superior performance can be attributed to the inherent strengths of our method, which effectively enhances the distinctions in anomalous samples by utilizing GIB-boosted losses with ID-label guidance. Moreover, we observe that GOODAT exhibits significant advantages in anomaly detection over the use of GTrans. This observation indicates the universality of GOODAT in contrast to other test-time-oriented techniques.

Ablation Study

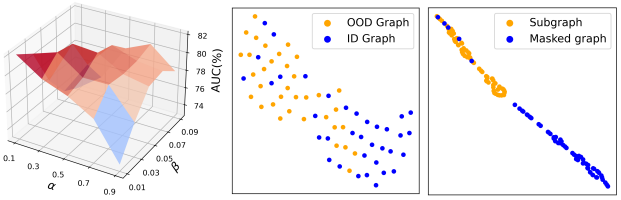
GOODAT incorporates subgraph GIB loss $\mathcal{L}_s$, masked graph GIB loss $\mathcal{L}_m$, and graph distribution separation loss $\mathcal{L}_d$. To evaluate the effectiveness of each of them, an ablation study is conducted and the results are summarized in Table 3, where a '✓' indicates the presence and a '–' denotes the absence of a component. Several key observations emerge from analyzing the table. (1) Utilizing all components concurrently yields optimal results on 8 out of 10 datasets, with decent results on the remaining 2 datasets. This demonstrates the effectiveness of integrating multiple loss functions to enhance graph OOD detection. (2) The distinct contributions of each loss function underscore their effectiveness as individual components. (3) Combining the two loss

Method	WL-OCSVM	WL-iF	InfoGraph-iF	GraphCL-iF	GTrans	Ours	*Improve**
PROTEINS-full	51.35±4.35	61.36±2.54	57.47±3.03	60.18±2.53	60.16±5.06	**77.92±2.37**	28.37%
AIDS	50.12±3.43	61.13±0.71	70.19±5.03	79.72±3.98	84.57±1.91	**95.50±0.99**	12.92%
DHFR	50.24±3.13	50.29±2.77	52.68±3.21	51.10±2.35	61.15±2.87	**61.52±2.86**	0.60%
BZR	50.56±5.87	52.46±3.30	63.31±8.52	60.24±5.37	51.97±8.15	**64.77±3.87**	2.31%
COX2	49.86±7.43	50.27±0.34	53.36±8.86	52.01±3.17	53.56±3.47	**59.99±9.76**	12.01%
DD	47.99±4.09	70.31±1.09	55.80±1.77	59.32±3.92	76.73±2.83	**77.62±2.88**	1.16%
IMDB-B	54.08±5.19	50.20±0.40	56.50±3.58	56.50±4.90	45.34±3.75	**65.46±4.34**	15.86%
REDDIT-B	49.31±2.33	48.26±0.32	68.50±5.56	71.80±4.38	69.71±2.21	**80.31±0.85**	11.85%
HSE	62.72±10.13	53.02±5.12	53.56±3.98	51.18±2.71	58.49±2.68	**63.05±0.90**	0.53%
MMP	55.24±3.26	52.68±3.34	54.59±2.01	54.54±1.86	48.19±3.74	**69.41±0.04**	25.65%
p53	54.59±4.46	50.85±2.16	52.66±1.95	53.29±2.32	53.74±2.98	**63.27±0.04**	15.90%
PPAR-gamma	57.91±6.13	49.60±0.22	51.40±2.53	50.30±1.56	56.20±1.57	**68.23±1.54**	17.82%
*Avg. Rank**	4.4	5	4.2	4.3	4.4	**2.1**	

Table 2: Anomaly detection results in terms of AUC score (%). *Full results are available at arXiv:2401.06176.

$\mathcal{L}_s$	$\mathcal{L}_m$	$\mathcal{L}_d$	ENZYMES	Tox21	FreeSolv	BBBP	ClinTox	Esol
			PROTEIN	SIDER	ToxCast	BACE	LIPO	MUV
✓	-	-	61.55±3.25	68.89±0.01	68.58±0.10	65.20±1.06	56.50±0.16	55.70±1.38
-	✓	-	52.74±0.01	68.87±0.01	67.65±0.08	63.77±0.24	66.71±0.05	85.82±0.08
-	-	✓	51.95±0.11	68.90±0.01	68.62±0.06	76.98±0.04	54.03±0.04	80.23±0.05
✓	✓	-	65.67±2.31	68.92±0.01	68.77±0.10	76.65±0.25	63.25±0.56	85.90±0.44
✓	-	✓	60.46±3.40	68.89±0.01	68.67±0.14	66.21±0.89	56.34±0.17	54.60±1.29
-	✓	✓	52.73±0.02	68.88±0.01	67.73±0.07	63.70±0.14	**66.76±0.08**	85.76±0.09
✓	✓	✓	**66.29±1.54**	**68.92±0.01**	**68.83±0.02**	**77.07±0.03**	62.46±0.54	**85.91±0.27**

Table 3: Ablation study results in terms of AUC score (%). Full results are available at arXiv:2401.06176.



(a) Influence on the parameter α and β. (b) OOD and ID G-Emb. (c) Sub and Masked G-Emb.

Figure 5: Parameter sensitivity analysis and visualization.

functions often leads to performance improvements, outperforming isolated evaluations of individual components.

Parameter Sensitivity Analysis

In GOODAT, two hyperparameters α (in Eq. 8) and β (in Eq. 10) are employed to control the involvement degree of subgraph GIB loss and masked graph GIB loss, respectively. We conduct a parameter sensitivity experiment on the PTC-MR/MUTAG dataset, where α is selected from $\{0.1, 0.3, 0.5, 0.7, 0.9\}$ and β is selected from $\{0.01, 0.03, 0.05, 0.07, 0.09\}$. As shown in Fig. 5 (a), when β is fixed, optimal outcomes are achieved with α in the range of 0.1-0.3. This implies that a slight level of compression on subgraphs proves the most effective results. Likewise, when α is held constant, β values in the range of 0.3-0.5 yield optimal results, which suggests that moderate compression of masked graph components enhances model effectiveness.

Visualization

To visually demonstrate the impact of our methods, we visualize distributions of subgraph embeddings to show the distinction between ID and OOD graphs. Fig. 5 (b) shows the embedding distributions of OOD subgraphs and ID subgraphs. We observe that the ID subgraph and OOD subgraph can be clearly distinguished, which indicates that GOODAT can detect OOD graphs intuitively. Fig. 5 (c) shows the distribution of subgraphs (i.e., Z) and masked graphs (i.e., Z'), where the distinction between subgraphs and masked graphs is also obvious. This demonstrates the effectiveness of our proposed graph distribution separation loss.

Conclusions

In this paper, we make the first attempt toward detecting graph out-of-distribution (OOD) samples at test time. To achieve this, we introduce a pioneering method, named GOODAT, which is a data-centric, unsupervised, and plug-and-play solution. With a graph masker applied to the input test graph, GOODAT identifies the clear differentiation between OOD graphs and ID graphs. We design three GIB-boosted losses to optimize the graph masker. Comprehensive experimentation demonstrates the superiority of GOODAT compared to baseline methods across diverse real-world benchmark datasets.

Acknowledgments

This work is supported by National Key Research and Development Program of China (2023YFC3304503), National Natural Science Foundation of China (92370111, 62276187, 62272340).

References

Alemi, A. A.; Fischer, I.; Dillon, J. V.; and Murphy, K. 2017. Deep Variational Information Bottleneck. In *ICLR 2017*.

Bai, H.; Canal, G.; Du, X.; Kwon, J.; Nowak, R. D.; and Li, Y. 2023. Feed Two Birds with One Scone: Exploiting Wild Data for Both Out-of-Distribution Generalization and Detection. In *ICML 2023*.

Breunig, M. M.; Kriegel, H.; Ng, R. T.; and Sander, J. 2000. LOF: Identifying Density-Based Local Outliers. In *SIGMOD 2000*.

Durante, F.; Fernandez-Sanchez, J.; and Sempi, C. 2013. A topological proof of Sklar's theorem. *Applied Mathematics Letters*, 26(9): 945–948.

Gui, S.; Li, X.; Wang, L.; and Ji, S. 2022. GOOD: A Graph Out-of-Distribution Benchmark. In *NeurIPS 2022*.

Guo, Y.; Yang, C.; Chen, Y.; Liu, J.; Shi, C.; and Du, J. 2023. A Data-centric Framework to Endow Graph Neural Networks with Out-Of-Distribution Detection Ability. In *KDD 2023*.

Hoffmann, M.; Galke, L.; and Scherp, A. 2023. Open-World Lifelong Graph Learning. In *IJCNN 2023*.

Huang, T.; Wang, D.; and Fang, Y. 2022. End-to-end open-set semi-supervised node classification with out-of-distribution detection. In *IJCAI 2022*.

Jin, D.; Wang, L.; Zhang, H.; Zheng, Y.; Ding, W.; Xia, F.; and Pan, S. 2023a. A survey on fairness-aware recommender systems. *Information Fusion*, 100: 101906.

Jin, D.; Wang, L.; Zheng, Y.; Li, X.; Jiang, F.; Lin, W.; and Pan, S. 2022. CGMN: A Contrastive Graph Matching Network for Self-Supervised Graph Similarity Learning. In *IJCAI 2022*.

Jin, D.; Wang, L.; Zheng, Y.; Song, G.; Jiang, F.; Li, X.; Lin, W.; and Pan, S. 2023b. Dual Intent Enhanced Graph Neural Network for Session-based New Item Recommendation. In *WWW 2023*.

Jin, W.; Zhao, T.; Ding, J.; Liu, Y.; Tang, J.; and Shah, N. 2023c. Empowering Graph Representation Learning with Test-Time Graph Transformation. In *ICLR 2023*.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. In *ICLR 2014*.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR 2017*.

Kullback, S.; and Leibler, R. A. 1951. On information and sufficiency. *The annals of mathematical statistics*, 22(1): 79–86.

Li, Z.; Wu, Q.; Nie, F.; and Yan, J. 2022. GraphDE: A Generative Framework for Debiased Learning and Out-of-Distribution Detection on Graphs. In *NeurIPS 2022*.

Liu, F. T.; Ting, K. M.; and Zhou, Z. 2008. Isolation Forest. In *ICDM 2008*.

Liu, Y.; Ding, K.; Liu, H.; and Pan, S. 2023a. GOOD-D: On Unsupervised Graph Out-Of-Distribution Detection. In *WSDM 2023*.

Liu, Y.; Ding, K.; Lu, Q.; Li, F.; Zhang, L. Y.; and Pan, S. 2023b. Towards Self-Interpretable Graph-Level Anomaly Detection. In *NeurIPS 2023*.

Liu, Y.; Ding, K.; Wang, J.; Lee, V.; Liu, H.; and Pan, S. 2023c. Learning Strong Graph Neural Networks with Weak Information. In *KDD 2023*.

Ma, R.; Pang, G.; Chen, L.; and van den Hengel, A. 2022. Deep Graph-level Anomaly Detection by Glocal Knowledge Distillation. In *WSDM 2022*.

Manevitz, L. M.; and Yousef, M. 2001. One-Class SVMs for Document Classification. *Journal of machine Learning research*, 2: 139–154.

Morris, C.; Kriege, N. M.; Bause, F.; Kersting, K.; Mutzel, P.; and Neumann, M. 2020. TUDataset: A collection of benchmark datasets for learning with graphs. In *ICML Workshop 2020*.

Neumann, M.; Garnett, R.; Bauckhage, C.; and Kersting, K. 2016. Propagation kernels: efficient graph kernels from propagated information. *Machine learning*, 102(2): 209–245.

Sehwag, V.; Chiang, M.; and Mittal, P. 2021. SSD: A Unified Framework for Self-Supervised Outlier Detection. In *ICLR 2021*.

Shervashidze, N.; Schweitzer, P.; van Leeuwen, E. J.; Mehlhorn, K.; and Borgwardt, K. M. 2011. Weisfeiler-Lehman Graph Kernels. *Journal of Machine Learning Research*, 12: 2539–2561.

Sklar, A. 1973. Random variables, joint distribution functions, and copulas. *Kybernetika*, 9(6): 449–460.

Sklar, M. 1959. Fonctions de répartition à n dimensions et leurs marges. In *Annales de l'ISUP*.

Song, Y.; and Wang, D. 2022. Learning on Graphs with Out-of-Distribution Nodes. In *KDD 2022*.

Stadler, M.; Charpentier, B.; Geisler, S.; Zügner, D.; and Günnemann, S. 2021. Graph posterior network: Bayesian predictive uncertainty for node classification. *NeurIPS 2021*, 34: 18033–18048.

Sun, F.; Hoffmann, J.; Verma, V.; and Tang, J. 2020. InfoGraph: Unsupervised and Semi-supervised Graph-Level Representation Learning via Mutual Information Maximization. In *ICLR 2020*.

Sun, Q.; Li, J.; Peng, H.; Wu, J.; Fu, X.; Ji, C.; and Yu, P. S. 2022. Graph Structure Learning with Variational Information Bottleneck. In *AAAI 2022*.

Tan, J.; Geng, S.; Fu, Z.; Ge, Y.; Xu, S.; Li, Y.; and Zhang, Y. 2022. Learning and Evaluating Graph Neural Network Explanations based on Counterfactual and Factual Reasoning. In *WWW 2022*.

Tan, Y.; Chen, C.; Zhuang, W.; Dong, X.; Lyu, L.; and Long, G. 2023a. Is heterogeneity notorious? taming heterogeneity

to handle test-time shift in federated learning. In *NeurIPS 2023*.

Tan, Y.; Liu, Y.; Long, G.; Jiang, J.; Lu, Q.; and Zhang, C. 2023b. Federated learning on non-iid graphs via structural knowledge sharing. In *AAAI 2023*.

Wang, L.; Hu, S.; Li, M.; and Zhou, J. 2019a. An exact algorithm for minimum vertex cover problem. *Mathematics*, 7(7): 603.

Wang, L.; Li, C.; Zhou, J.; Jin, B.; and Yin, M. 2019b. An Exact Algorithm for Minimum Weight Vertex Cover Problem in Large Graphs. *CoRR*, abs/1903.05948.

Wang, L.; Zheng, Y.; Jin, D.; Li, F.; Qiao, Y.; and Pan, S. 2023a. Contrastive Graph Similarity Networks. *ACM Transactions on the Web*.

Wang, X.; Dong, Y.; Jin, D.; Li, Y.; Wang, L.; and Dang, J. 2023b. Augmenting Affective Dependency Graph via Iterative Incongruity Graph Learning for Sarcasm Detection. In *AAAI 2023*.

Wu, B.; Zhang, H.; Yang, X.; Wang, S.; Xue, M.; Pan, S.; and Yuan, X. 2024. GraphGuard: Detecting and Counteracting Training Data Misuse in Graph Neural Networks. In *NDSS*.

Wu, J.; Li, C.-M.; Wang, L.; Hu, S.; Zhao, P.; and Yin, M. 2023a. On solving simplified diversified top-k s-plex problem. *Computers & Operations Research*, 153: 106187.

Wu, J.; Li, C.-M.; Zhou, Y.; Yin, M.; Xu, X.; and Niu, D. 2022. HEA-D: A Hybrid Evolutionary Algorithm for Diversified Top-k Weight Clique Search Problem. In *IJCAI 2022*. Main Track.

Wu, Q.; Chen, Y.; Yang, C.; and Yan, J. 2023b. Energy-based Out-of-Distribution Detection for Graph Neural Networks. In *ICLR 2023*.

Wu, T.; Ren, H.; Li, P.; and Leskovec, J. 2020. Graph information bottleneck. *NeurIPS 2020*, 33: 20437–20448.

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2019. How Powerful are Graph Neural Networks? In *ICLR 2019*.

You, Y.; Chen, T.; Sui, Y.; Chen, T.; Wang, Z.; and Shen, Y. 2020. Graph Contrastive Learning with Augmentations. In *NeurIPS 2020*.

Yu, J.; Xu, T.; Rong, Y.; Bian, Y.; Huang, J.; and He, R. 2021. Graph Information Bottleneck for Subgraph Recognition. In *ICLR 2021*.

Yu, Z.; Jin, D.; Wang, X.; Li, Y.; Wang, L.; and Dang, J. 2023. Commonsense Knowledge Enhanced Sentiment Dependency Graph for Sarcasm Detection. In *IJCAI 2023*.

Zhang, H.; Wu, B.; Wang, S.; Yang, X.; Xue, M.; Pan, S.; and Yuan, X. 2023a. Demystifying Uneven Vulnerability of Link Stealing Attacks against Graph Neural Networks. In *ICML*, volume 202 of *Proceedings of Machine Learning Research*, 41737–41752. PMLR.

Zhang, H.; Wu, B.; Yang, X.; Zhou, C.; Wang, S.; Yuan, X.; and Pan, S. 2021. Projective Ranking: A Transferable Evasion Attack Method on Graph Neural Networks. In *CIKM*, 3617–3621. ACM.

Zhang, H.; Wu, B.; Yuan, X.; Pan, S.; Tong, H.; and Pei, J. 2022. Trustworthy Graph Neural Networks: Aspects, Methods and Trends. *CoRR*, abs/2205.07424.

Zhang, H.; Yuan, X.; Nguyen, Q. V. H.; and Pan, S. 2023b. On the Interaction between Node Fairness and Edge Privacy in Graph Neural Networks. *CoRR*, abs/2301.12951.

Zhang, H.; Yuan, X.; Zhou, C.; and Pan, S. 2023c. Projective Ranking-Based GNN Evasion Attacks. *IEEE Trans. Knowl. Data Eng.*, 35(8): 8402–8416.

Zhao, X.; Chen, F.; Hu, S.; and Cho, J. 2020. Uncertainty Aware Semi-Supervised Learning on Graph Data. In *NeurIPS 2020*.

Zheng, X.; Liu, Y.; Bao, Z.; Fang, M.; Hu, X.; Liew, A. W.-C.; and Pan, S. 2023a. Towards Data-centric Graph Machine Learning: Review and Outlook. *arXiv preprint arXiv:2309.10979*.

Zheng, X.; Zhang, M.; Chen, C.; Molaei, S.; Zhou, C.; and Pan, S. 2023b. GNNEvaluator: Evaluating GNN Performance On Unseen Graphs Without Labels. *NeurIPS 2023*.

Zheng, Y.; Koh, H. Y.; Ju, J.; Nguyen, A. T.; May, L. T.; Webb, G. I.; and Pan, S. 2023c. Large language models for scientific synthesis, inference and explanation. *arXiv preprint arXiv:2310.07984*.

Zheng, Y.; Lee, V. C.; Wu, Z.; and Pan, S. 2021. Heterogeneous graph attention network for small and medium-sized enterprises bankruptcy prediction. In *PAKDD 2021*.

Zheng, Y.; Pan, S.; Lee, V.; Zheng, Y.; and Yu, P. S. 2022. Rethinking and scaling up graph contrastive learning: An extremely efficient approach with group discrimination. *NeurIPS 2022*.

Zheng, Y.; Zhang, H.; Lee, V. C.; Zheng, Y.; Wang, X.; and Pan, S. 2023d. Finding the Missing-half: Graph Complementary Learning for Homophily-prone and Heterophily-prone Graphs. In *ICML 2023*.