Learning Node Abnormality with Weak Supervision

Qinghai Zhou
University of Illinois at Urbana-Champaign
qinghai2@illinois.edu

Kaize Ding
Northwestern University
kaize.ding@northwestern.edu

Huan Liu
Arizona State University
huanliu@asu.edu

Hanghang Tong
University of Illinois at Urbana-Champaign
htong@illinois.edu

ABSTRACT

Graph anomaly detection aims to identify the atypical substructures and has attracted an increasing amount of research attention due to its profound impacts in a variety of application domains, including social network analysis, security, finance, and many more. The lack of prior knowledge of the ground-truth anomaly has been a major obstacle in acquiring fine-grained annotations (e.g., anomalous nodes), therefore, a plethora of existing methods have been developed either with a limited number of node-level supervision or in an unsupervised manner. Nonetheless, annotations for coarse-grained graph elements (e.g., a suspicious group of nodes), which often require marginal human effort in terms of time and expertise, are comparatively easier to obtain. Therefore, it is appealing to investigate anomaly detection in a weakly-supervised setting and to establish the intrinsic relationship between annotations at different levels of granularity. In this paper, we tackle the challenging problem of weakly-supervised graph anomaly detection with coarse-grained supervision by (1) proposing a novel architecture of graph neural network with attention mechanism named WEDGE that can identify the critical node-level anomaly given a few labels of anomalous subgraphs, and (2) designing a novel objective with contrastive loss that facilitates node representation learning by enforcing distinctive representations between normal and abnormal graph elements. Through extensive evaluations on real-world datasets, we corroborate the efficacy of our proposed method, improving AUC-ROC by up to 16.48% compared to the best competitor.

CCS CONCEPTS

• **Information systems** → **Data mining**.

KEYWORDS

Graph anomaly detection; Weak supervision

ACM Reference Format:
Qinghai Zhou, Kaize Ding, Huan Liu, and Hanghang Tong. 2023. Learning Node Abnormality with Weak Supervision. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3583780.3614950

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

ACM ISBN 979-8-4007-0124-5/23/10…$15.00
https://doi.org/10.1145/3583780.3614950

1 INTRODUCTION

Graph-structured data is ubiquitous in various real-world scenarios, including item co-purchase graphs [52] in e-commerce, social networks [62] on social media platforms, and molecular graphs [60] in drug design. To harness the rich information encoded in graph-structured data, a variety of graph analytical tasks have been studied in recent years, such as node classification [25, 57], graph classification [54, 70], network alignment [58, 72] and many more [16, 17, 59]. Among others, graph anomaly detection has received much attention due to its profound impacts in different applications such as fraud detection [14] and social spam detection [39]. Essentially, this task aims to detect the instances that significantly deviate from the majority of instances.

Obtaining a large amount of annotated data often requires laborious labeling costs and intensive domain knowledge [10, 11, 31, 71]. Therefore, many efforts have been made either with limited node-level supervision or in an unsupervised fashion. Though great success has been made, existing methods still suffer from the lack of fine-grained supervision signals and could fail to perform accurate detection. Compared with node-level labels, subgraph-level supervision requires much less effort to obtain in many applications. For example, in financial fraud detection, it is relatively easier to determine the presence of suspicious money laundering activities within a group of users; nonetheless, accurately identifying the actual fraudulent users is notably much more difficult due to their complicated disguise [36, 64]. In disaster management, it is hard, if not impossible, to pinpoint the epicenter (node-level anomaly) in the immediate aftermath of a natural disaster (e.g., hurricane); on the other hand, we can often roughly locate the impacted communities (subgraph-level anomaly) to support rapid rescue [4, 7]. During the criminal investigation, law-enforcement often first identifies a group of suspects/persons of interest (subgraph-level anomaly), before capturing the master criminal mind (node-level anomaly). We ask: *how can we improve the node-level anomaly detection under such a weakly supervised setting where coarse-grained group labels are available?*

However, it is a highly non-trivial task to leverage such coarse-grained weak supervision signals [12] for node-level graph anomaly detection, mainly due to the following reasons. First, under the weakly-supervised setting, the graph anomaly detector can only access coarse-grained labels for subgraphs that contains a group of nodes, while our objective is to detect node-level anomalies, i.e., abnormal nodes. Though previous works [2, 48, 74] in multiple

instance learning (MIL) have been proposed to solve similar problems, those methods either fail to capture the data heterogeneity in graphs or only focus on group or bag-level prediction. Thus directly applying existing MIL methods to our problem would inevitably result in sub-optimal results or even become infeasible. Second, though coarse-grained labels are relatively easier to be accessed, such weak supervision signals are often noisy and may hamper the model performance if we directly take them as ground-truth labels without appropriate treatments [30]. Therefore, how to mitigate the detrimental effects of noisy-labeled data and learn expressive node representations to further distinguish abnormal nodes from normal ones is another challenge to be solved.

To address the aforementioned challenges, in this paper, we propose a novel architecture, namely WEDGE, for detecting node-level graph anomalies with weak supervision of coarse-grained subgraph-level labels. The key innovation of WEDGE lies in that we effectively establish quantitative relationship between nodes and subgraphs in terms of both their representations and the predicted abnormalities. To be specific, WEDGE first incorporates a GNN-based node-level abnormality predictor to learn the representation and estimate the anomaly score for each node. Then, WEDGE leverages a subgrah abnormality predictor equipped with attention mechanism to quantify the importance of each node in the subgraph, and to compute the subgraph-level representations and anomaly scores via a weighted aggregation. To guide the model training, we adopt a deviation loss [38] defined on subgraph anomaly scores, which enforces a momentous deviation between the anomaly scores of abnormal subgraphs/nodes and that of normal subgraphs/nodes. To further enhance the representation learning of subgraphs/nodes, we propose a contrastive objective for maximizing the closeness between subgraphs with similar abnormality in the embedding space. The seamlessly integrated WEDGE framework empowers the representation learning directly optimized for node-level graph anomaly detection, which in turn significantly mitigates the limitations of high cost in obtaining fine-grained annotations. We summarize the main contributions of this paper as follows:

- **Problem:** We formally define the problem of weakly-supervised graph anomaly detection. The key idea is to extract critical knowledge regarding node abnormality from subgraphs with coarse-grained labels.
- **Algorithms:** we propose a novel framework WEDGE, which consists of a GNN-based architecture equipped with attention mechanism, and objectives directly optimized for graph anomaly detection with weak supervision.
- **Experiments:** we conduct extensive evaluations on real-world datasets to demonstrate the superiority of the proposed WEDGE, which outperforms the best competitor by up to 16.48% in terms of AUC-ROC.

2 PROBLEM DEFINITION

In this section, we formally define the problem of weakly-supervised graph anomaly detection, after the notations are introduced.

2.1 Notations

Throughout the paper, we use bold uppercase letters for matrices (e.g., $\mathbf{A}$), bold lowercase letters for vectors (e.g., $\mathbf{h}$), calligraphy letters for sets (e.g., $\mathcal{V}$) and lowercase letters for scalars (e.g., k). In this work, we focus on a node-attributed graph, i.e., $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$, where $\mathcal{V}$ is the set of nodes, i.e., $\{v_1, v_2, \ldots, v_n\}$, $\mathcal{E}$ represents the set of edges, i.e., $\{e_1, e_2, \ldots, e_m\}$. We use $\mathbf{X} = [\mathbf{x}_1^\mathrm{T}, \mathbf{x}_2^\mathrm{T}, \cdots, \mathbf{x}_n^\mathrm{T}] \in \mathbb{R}^{n \times d}$ to denote the node attributes matrix, where $\mathbf{x}_i$ is the attribute vector for node v_i. Alternatively, we represent the attributed graph as $\mathcal{G} = (\mathbf{A}, \mathbf{X})$, where $\mathbf{A} = \{0, 1\}^{n \times n}$ is the adjacency matrix representing the graph topology. To be specific, $\mathbf{A}_{i,j} = 1$ implies that there is an edge between node v_i and node v_j, otherwise, $\mathbf{A}_{i,j} = 0$.

2.2 Problem Definition

In weakly supervised graph anomaly detection, we are only provided with the labels for the entire subgraph. Considering that an abnormal subgraph may contain both normal and abnormal nodes, the coarse-grained subgraph labels are not precise to be directly applied to infer node-level abnormality.

Multiple instance learning (MIL), which deals with training data organized in sets (bags) where bag-level supervision is provided, is similar to the problem of weakly supervised graph anomaly detection. Nonetheless, our problem is different from the classic MIL in the following two aspects, including (1) compared to the i.i.d. data assumption, graph anomaly detection aims to capture the structural correlation between nodes, and (2) we aim to infer node-level anomalies, which is more challenging than standard bag-level prediction. let us first briefly review the background of multiple instance learning (MIL) [74].

In MIL, a *bag* is defined as a group of individual training *instances* where the label of each instance is unknown. One bag is labeled as positive if this bag contains at least one positive instance and negative otherwise. Given a bag of b instances, i.e., $\mathcal{B} = \{x_1, x_2, \ldots, x_b\}$, MIL aims to predict the bag-level label as follows [22],

$$y_{\mathcal{B}} = f(g(x_1), g(x_2), \ldots, g(x_b)) \tag{1}$$

where $g(\cdot)$ is an instance-level transformation to predict instance labels or to produce feature representations, and $f(\cdot)$ functions as an aggregator to produce the final bag-level prediction according to the node-level outcome from $g(\cdot)$.

In our setting, a subgraph is considered as a bag and the node-level (i.e., instance-level) labels are unknown. We assume to have a set of k subgraphs (bags) that are labeled as anomalous, i.e., $\mathcal{B}_a = \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_k\}$. The set of nodes in $\mathcal{B}_a$ is denoted as $\mathcal{V}^l$ and the set of remaining unlabeled nodes is represented as $\mathcal{V}^u$. Note that $\mathcal{V} = \{\mathcal{V}^l, \mathcal{V}^u\}$ and in our problem $|\mathcal{V}^l| \ll |\mathcal{V}^u|$ since only a limited number of labeled subgraphs are given. For each subgraph (e.g., $\mathcal{S}_i$), it contains multiple nodes from the graph $\mathcal{G}$, i.e., $\mathcal{S}_i = \{v_1^{(i)}, v_2^{(i)}, \ldots, v_{b_i}^{(i)}\}$ where b_i is the size of subgraph $\mathcal{S}_i$. A positive label (i.e., $Y = 1$) is associated with a subgraph if there exists at least one anomalous node in this subgraph, and $Y = 0$ otherwise. Formally, we have the following definition of abnormal subgraph.

Definition 1. Abnormal Subgraph. Given an attributed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$, a connected subgraph $\mathcal{S}$ of graph $\mathcal{G}$ is defined as abnormal if it contains at least one anomalous node.

Generally speaking, the goal of weakly-supervised graph anomaly detection is to maximally improve the accuracy of detecting node-level anomalies on the graph by effectively leveraging the limited knowledge of coarse-grained annotations for anomalous subgraphs. Following the convention of graph anomaly detection [1], we formulate the problem of weakly-supervised graph anomaly detection as a ranking problem, and give the formal definition as:

Problem 1. ***Weakly-supervised Graph Anomaly Detection***

Given: *a node-attributed graph* $\mathcal{G} = (\mathbf{A}, \mathbf{X})$ *which contains a set of* k *labeled anomalous subgraphs (i.e.,* $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_k$*).*

Find: *a model for node-level anomaly detection, which is capable of leveraging the knowledge of coarse-grained ground-truth (i.e.,* $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_k$*), to detect the abnormal nodes in the graph* $\mathcal{G}$*. Ideally, detected anomalies should have higher ranking scores than that of the normal nodes.*

3 PROPOSED APPROACH

In this section, we present the details of the proposed framework, WEDGE, for weakly-supervised graph anomaly detection. The key innovation of WEDGE lies in that we precisely quantify the intrinsic relationship between nodes and subgraphs from the perspectives of both abnormalities and embeddings. To be specific, in terms of abnormality, we propose an end-to-end framework, which incorporates graph neural networks (GNNs) and attention mechanism, to facilitate the node-level anomaly detection on graph with limited, coarse-grained labeled subgraphs. Additionally, we adopt an objective integrating subgraph-level supervision and self-supervised contrastive loss, being able to establish the quantitative correlation between nodes and subgraphs in terms of their embeddings. We illustrate the overview of the proposed framework in Figure 1.

3.1 Framework for Weakly-supervised Graph Anomaly Detection

In weakly-supervised anomaly detection, we aim to leverage the coarse-grained subgraph-level labels to enable the fine-grained node-level anomaly detection, and each anomalous node/subgraph is anticipated to be assigned a large score representing its high level of abnormality. To achieve this, we propose a multiple instance learning framework, named WEDGE, that includes the following key components: (1) a *node abnormality predictor* as the node-level extractor (i.e., $g(\cdot)$ in Eq. (1)), and (2) a novel *subgraph abnormality predictor* as the aggregation function (i.e., $f(\cdot)$ in Eq. (1)). Essentially, the *node abnormality predictor* is composed of a *graph encoder* for learning the node representations, concatenated with a *score estimator* module for computing the anomaly score for each node. Afterwards, the obtained node representations and anomaly scores will be forwarded to the *subgraph abnormality predictor* for estimating the overall anomaly scores of the subgraphs. We detail the proposed approach as follows.

Node Abnormality Predictor $g(\cdot)$**.** In order to evaluate the level of abnormality for each node, we propose the *node abnormality predictor* module to assign an anomaly score for each node from the graph $\mathcal{G}$. This module is composed of two sub-components, including (1) a *graph encoder*, and (2) a *score estimator*.

(1) *Graph Encoder.* Informative node representations serve as the cornerstone for node anomaly detection. To construct a high-quality *graph encoder* module, specifically, we exploit GNNs to map each node to a low-dimensional latent space. GNNs define a general architecture of neural network on graph-structured data. This architecture can capture the local graph structure as well as features of nodes following the neighborhood message-passing mechanism. The intermediate node representations can be obtained as follows:

$$\mathbf{h}_i^l = \sigma\Big(\mathbf{h}_i^{l-1} + \textsc{Aggregate}^l(\{\mathbf{h}_j^{l-1} | \forall j \in \mathcal{N}_i \cup \{v_i\}\})\Big) \quad (2)$$

where $\mathbf{h}_i^l$ is the intermediate representation of node v_i at the l-th layer, $\mathcal{N}_i$ is the set of one-hop neighboring nodes of node v_i. Particularly, $\textsc{Aggregate}^l(\cdot)$ is a function that integrates information from the neighboring nodes including v_i itself. $\sigma(\cdot)$ denotes the nonlinear activation (e.g., ReLU).

The final node representations can be obtained by applying the information aggregation procedure in an iterative manner. We use $\mathbf{Z} = [\mathbf{z}_1^\mathrm{T}, \mathbf{z}_2^\mathrm{T}, \ldots, \mathbf{z}_n^\mathrm{T}] \in \mathbb{R}^{n \times d_g}$ to denote the learned representations for all the nodes from the GNNs. It is worth noting that the *graph encoder* is compatible with arbitrary GNN-based architecture [23, 25, 49, 53], and here we apply Graph Convolutional Networks (GCNs) [25] in our implementation.

Then, the *graph encoder* transforms the obtained node representations (i.e., $\mathbf{z}_1, \ldots, \mathbf{z}_n$) from GNNs to another latent space through a nonlinear activation. Concretely, the transformation can be achieved by a one-layered feed-forward neural network as follows:

$$\mathbf{Q} = \sigma(\mathbf{Z}\mathbf{W}_{e_1}), \quad (3)$$

where $\mathbf{Q} \in \mathbb{R}^{n \times d_e}$ is the final representation matrix of all nodes, $\mathbf{W}_{e_1}$ is the learnable weight matrix, and $\sigma(\cdot)$ represents the nonlinear ReLU activation. In practice, we observe that such non-linear transformation can improve the detection performance compared to directly utilizing the representations from the GNNs. For simplicity, we denote the *graph encoder* as g_{θ_g}.

(2) *Score Estimator.* The *score estimator* computes a real-valued anomaly score for each node based on the final representations (i.e., $\mathbf{Q}$) as follows:

$$\mathbf{c} = \mathbf{Q}\mathbf{w}_{e_2}, \quad (4)$$

where $\mathbf{c} \in \mathbb{R}^{n \times 1}$ represents the anomaly score vector and $\mathbf{w}_{e_2} \in \mathbb{R}^{d_e \times 1}$ is the learnable weight vector. The bias terms are omitted. We use a parameterized function $g_{\theta_e}(\cdot)$ to denote the *score estimator* hence the *node abnormality predictor* can be represented by $g_{\theta_n}(\mathbf{A}, \mathbf{X}) = g_{\theta_e}(g_{\theta_g}(\mathbf{A}, \mathbf{X}))$.

Subgraph Abnormality Predictor $f(\cdot)$**.** The subgraph abnormality is evaluated from the following two intuitive perspectives, including (1) each node in the subgraph reveals a level of abnormality, ranging from extremely low (i.e., normal) to extremely high (i.e., highly anomalous), which can be estimated by the *node abnormality predictor*, and (2) nodes indicate different degrees of importance to the overall abnormality of the subgraph. Hence, the overall subgraph-level abnormality is considered as an aggregation of node abnormalities, weighted by the corresponding node importance. Therefore, our proposed *subgraph abnormality predictor* aims to (1) differentiate the critical nodes by accurately evaluating the importance of nodes, and (2) estimate the overall subgraph abnormality based on the resulting importance. In essence, the *subgraph abnormality predictor* consists of two key modules: (1) a *significance evaluator* based on attention mechanism for allocating importance to nodes inside a subgraph, and (2) an *aggregator* for computing the anomaly score of the subgraph and updating the subgraph representation according to the resulting attention weights. We represent the *subgraph abnormality predictor* as a parameterized function $f_{\theta_s}(\cdot)$. The detailed description is as follows.

(1) *Significance Evaluator.* Recall that in weakly-supervised graph anomaly detection, we have a set of labeled anomalous subgraphs, i.e., $\mathcal{B} = \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_k\}$. For each subgraph $\mathcal{S}_i \in \mathcal{B}$, the goal of the *significance evaluator* is to estimate the contribution of each individual node to the overall subgraph abnormality. Specifically, we first compute the attention vector (i.e., $\mathbf{p}_j$) for node $v_j^{(i)}$ through an one-layered feed-forward network,

$$\mathbf{p}_j = \tanh\left(\mathbf{W}_p \mathbf{q}_j + \mathbf{b}_p\right) \quad (5)$$

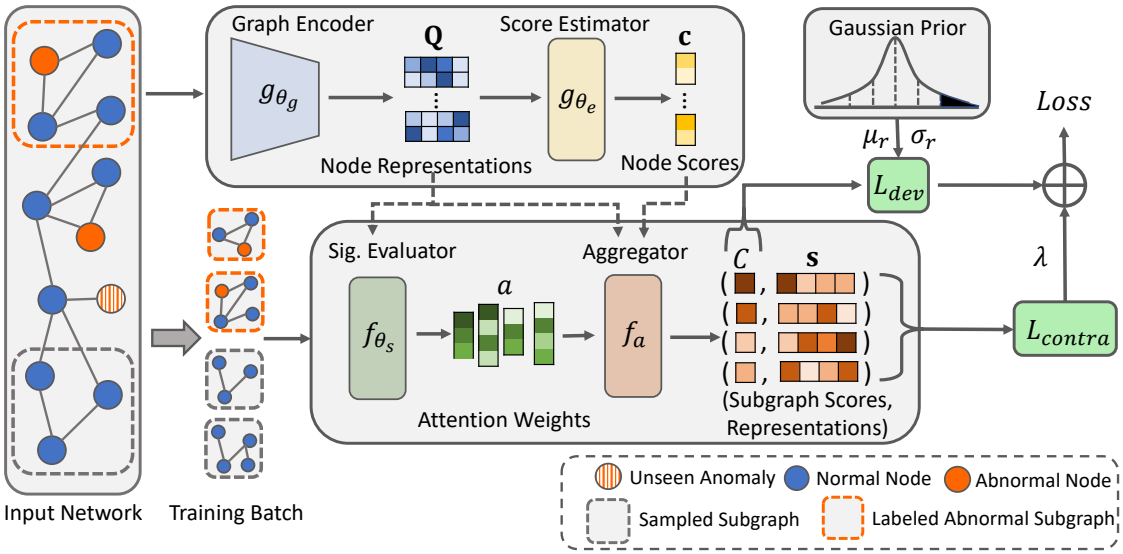

Figure 1: The illustration of the proposed WEDGE framework for weakly-supervised graph anomaly detection. WEDGE is trained for node-level graph anomaly detection with coarse-grained subgraph-level supervision. The key is to establish the intrinsic correspondence between nodes and subgraphs w.r.t. abnormality and embeddings. Best viewed in color.

where $\mathbf{q}_j$ is the representation of node $v_j^{(i)}$ obtained from the *node abnormality predictor*, $\mathbf{W}_p$ and $\mathbf{b}_p$ are the parameter matrix and the bias vector, respectively. $\tanh(\cdot)$ represents the element-wise hyperbolic tangent function.

It is then straightforward to calculate the attention weight for node $v_j^{(i)}$ as:

$$a_j = \frac{\exp(\mathbf{p}_j^{\mathrm{T}}\mathbf{w}_a)}{\sum_{v_m^{(i)} \in \mathcal{S}_i} \exp(\mathbf{p}_m^{\mathrm{T}}\mathbf{w}_a)} \tag{6}$$

where Eq. (6) defines the normalized similarity between the attention vector $\mathbf{p}_j$ and the learnable vector $\mathbf{w}_a$. Intuitively, vector $\mathbf{w}_a$ is crucial in attention mechanism and is able to recognize critical anomaly nodes. We denote the *significance evaluator* as $f_{\theta_i}(\cdot)$.

(2) *Aggregator.* Having obtained the normalized weights for each node in the subgraph, we can proceed to compute anomaly score for each subgraph as the weighted sum of node-level anomaly scores,

$$C_i = \sum_{v_j^{(i)} \in \mathcal{S}_i} a_j c_j, \tag{7}$$

where c_j is the anomaly score of node $v_j^{(i)}$ from the *node abnormality predictor*.

Similarly, we can obtain the representation of the subgraph $\mathcal{S}_i$ as follows:

$$\mathbf{s}_i = \sum_{v_j^{(i)} \in \mathcal{S}_i} a_j \mathbf{q}_j. \tag{8}$$

We denote the *aggregator* as $f_a(\cdot)$ hence the *subgraph abnormality predictor* can be represented by $f_{\theta_s}(\cdot) = f_a(f_{\theta_i}(\cdot))$, which, in essence, transforms the node-level representations/scores to the subgraph-level counterparts. The entire WEDGE model can be concretely represented as $f_\theta(\mathbf{A}, \mathbf{X}) = f_{\theta_s}(g_{\theta_n}(\mathbf{A}, \mathbf{X}))$ and directly maps the input graph to subgraph representations/anomaly scores, therefore can be trained in an end-to-end fashion.

3.2 Training

The objective of our proposed WEDGE framework is to discriminate normal and abnormal nodes/subgraphs based on the computed anomaly scores from the *node/subgraph abnormality predictor*. To navigate the model training with limited subgraph-level supervision, inspired by [38], we propose a subgraph-level deviation loss to enforce the model to allocate significantly larger anomaly scores to true anomalous nodes/subgraphs whose patterns deviate from the normal ones to a great extent. Considering that we are provided with very limited supervision signals, we further design a contrastive loss to enhance the representation learning. The key idea of the contrastive loss is to maximize (1) the similarity between anomalous subgraph representations, and (2) the disparity between normal and abnormal subgraphs. The details of learning objectives are as follows.

Subgraph-level Deviation Loss for Graph Anomaly Detection. For a subgraph $\mathcal{S}_i$, the deviation is defined as the distance between the anomaly score (i.e., C_i) and the *reference score* in the format of standard score: $\mathrm{dev}(\mathcal{S}_i) = \frac{C_i - \mu_r}{\sigma_r}$, where *reference score*, i.e., μ_r, is the mean value of r anomaly scores sampled from a Gaussian distribution (i.e., $\{s_1, \ldots, s_r\} \sim \mathcal{N}(\mu, \sigma^2)$) [27, 38] and σ_r is the corresponding standard deviation. The objective function for deviation loss is derived as follows:

$$\mathcal{L}_{\mathrm{dev}} = (1 - Y_i) \cdot |\mathrm{dev}(\mathcal{S}_i)| + Y_i \cdot \max(0, m - \mathrm{dev}(\mathcal{S}_i)), \tag{9}$$

where Y_i is the ground-truth label of the subgraph $\mathcal{S}_i$. m is a confidence margin which defines a radius within the deviation. In practice, we choose m to be a large value (e.g., $m = 5$) to ensure it is larger than the deviation.

Through minimizing Eq. (9), the *abnormality predictor* will enforce a large positive deviation between the anomaly score of an anomalous subgraph and the *reference score* μ_r, while confining the anomaly scores of normal subgraphs around μ_r. Since the subgraph-level representations and anomaly scores are directly influenced by the node-level counterparts, the deviation loss can further improves the representation learning of nodes for anomaly detection.

Contrastive Self-supervision. In general, self-supervised contrastive learning aims to empower the representation learning by maximizing the agreement between similar instances in each

instance pair while capturing the negative correlation between mismatching patterns [5]. In weakly-supervised graph anomaly detection, we are only provided with limited number of labeled anomalous subgraphs. To address this challenge, we propose a contrastive loss to further refine the representations learned from the *node/subgraph abnormality predictor*, through which the embeddings of nodes and subgraphs can be quantitatively correlated. The intuition is as follows, by contrasting two anomalous subgraphs, the corresponding similarity is expected to be larger than that when comparing two subgraphs in two categories.

Specifically, we randomly sample k subgraphs with the similar size from the remaining network and treat them as normal subgraphs. Note that we also use the sampled subgraphs as the negative instances to compute the deviation loss in Eq. (9). It is also worth mentioning that since the sampled subgraphs may contain both normal nodes and unlabeled abnormal nodes, hence contamination is introduced to the training set. The experimental results demonstrate that our proposed framework consistently performs well with this simple sampling strategy and is robust to various levels of contamination. We present the robustness analysis on contamination level in Sec. 4.4.

For each training epoch i, we select N labeled anomalous subgraphs and N sampled subgraphs without replacement, forming a batch of size $2N$. We denote the training batch as $\mathcal{B}_i = \mathcal{B}_a^{(i)} \cup \mathcal{B}_n^{(i)}$ where $\mathcal{B}_a^{(i)}$ and $\mathcal{B}_n^{(i)}$ represent the selected labeled subgraphs and sampled subgraphs at the i-th epoch, respectively. We define the positive pair as a combination any two different subgraphs from $\mathcal{B}_a^{(i)}$. A negative pair is composed of one subgraph from $\mathcal{B}_a^{(i)}$ and the other one from $\mathcal{B}_n^{(i)}$. To compute the loss for a positive pair of subgraphs $(\mathcal{S}_i, \mathcal{S}_j)$, we have,

$$l_{\mathcal{S}_i,\mathcal{S}_j} = -\log \frac{\exp(\text{sim}(\mathbf{s}_i, \mathbf{s}_j)/\tau)}{\sum_{\mathcal{S}_k \in \mathcal{B}_n \cup \{\mathcal{S}_j\}} \exp(\text{sim}(\mathbf{s}_i, \mathbf{s}_k)/\tau)}, \tag{10}$$

where $\mathbf{s}_i$ is the representation of $\mathcal{S}_i$ and $\text{sim}(\cdot,\cdot)$ represents the cosine similarity between two vectors, and $\tau > 0$ is a temperature parameter. Eq. (10)

We then obtain the final contrastive loss by computing l over all positive pairs:

$$\mathcal{L}_{\text{contra}} = \frac{1}{N(N-1)} \sum_{\mathcal{S}_i,\mathcal{S}_j \in \mathcal{B}_a} \left[l_{\mathcal{S}_i,\mathcal{S}_j} + l_{\mathcal{S}_j,\mathcal{S}_i} \right]. \tag{11}$$

Through minimizing Eq. (11), the model is able to enforce (1) the closeness between the representations of anomalous subgraphs, and (2) the disparity between the abnormal and normal subgraphs in the embedding space.

The deviation loss in Eq. (9) and the contrastive loss in Eq. (11) are mutually complementary to each other in the following way. From the perspective of subgraph representations, the contrastive loss empowers the learning by contrasting every pair of subgraphs in the same training batch, which enables the inter-subgraph correlation to be captured. In the mean while, the deviation loss determines the detection outcome (i.e., anomaly scores) for evaluation and enforces a significantly higher score being assigned to an abnormal node/subgraph based on the learned representations.

Therefore, we combine the two objectives as follows:

$$\mathcal{L} = \mathcal{L}_{\text{dev}} + \lambda \cdot \mathcal{L}_{\text{contra}}, \tag{12}$$

where λ is a regularization parameter. It is worth mentioning that if the node-level supervision is also available, our proposed WEDGE framework can readily ingest such fine-grained supervision, which makes WEDGE applicable to a wider range of scenarios.

We summarize the full algorithm in Algorithm 1.

Algorithm 1 The learning algorithm of WEDGE

Input: (1) input network $\mathcal{G} = (\mathbf{A}, \mathbf{X})$; (2) a set of k labeled subgraphs $\mathcal{B}_a = \{\mathcal{S}_1, \ldots, \mathcal{S}_k\}$; (3) training epochs E, batch size $2N$, hyper-parameters τ and λ.
Output: Anomaly scores of nodes in $\mathcal{V}^u$.

```
Initialize model parameters;
Construct B_n by sampling k subgraphs from V^u;
while e < E do
    Randomly sample N subgraphs from B_a and N from B_n to comprise the batch B_e;
    Compute node-level representations and anomaly scores using Eq. (3) and (4), respectively;
    Compute subgraph-level representations and anomaly scores using Eq. (8) and (7), respectively;
    Compute the loss in Eq. (12);
    Back-propagate the loss and update model parameters;
end while
Compute the anomaly scores for nodes in V^u;
```

4 EXPERIMENTS

In this section, we conduct the empirical evaluations to demonstrate the effectiveness of the proposed framework and we aim to answer the following research questions:

- **RQ1.** How effective is the proposed WEDGE framework in detecting node-level anomalies with coarse-grained subgraph labels?
- **RQ2.** How does each component of the proposed WEDGE framework (i.e., attention mechanism, contrastive loss) contribute to the detection performance?
- **RQ3.** How robust is WEDGE to different levels of contamination and how sensitive is WEDGE to the model parameters?

4.1 Experimental Setup

Evaluation Datasets. We use four real-world datasets, including *Yelp*, *Amazon*, *PubMed* and *Reddit*, which are publicly available and have been widely adopted in previous research [19, 25, 39, 42]. Table 1 summarizes the statistics of each dataset. The detailed description is as follows.

- **Yelp** [39] is collected from Yelp.com and contains reviews for restaurants located in New York City. The reviewers are classified into two classes, abnormal (reviewers with only filtered reviews) and normal (reviewers with no filtered reviews) according to the Yelp anti-fraud filtering algorithm. We select a subset of total reviews and construct the network as follows: nodes represent reviewers and there is a link between two reviewers if they have commented on the same restaurant. We apply the bag-of-words model [65] on the textual contents to obtain the node attributes.
- **Amazon** [34] contains the product review information from Amazon under the category of office products. Following [63], reviewers with more than 80% helpful votes are labeled as normal and abnormal otherwise. We construct the review graph by connecting the reviewers that have commented on the same product

and extract bag-of-words features from review content [65] as the node attributes.

- **PubMed** [42] is a citation network where nodes represent medical articles related to diabetes and edges are citations relations. Node attribute is represented by a TF/IDF weighted word vector from a dictionary which consists of 500 unique words.
- **Reddit** [19] is collected from reddit.com, an online discussion forum, where nodes represent threads and an edge exists between two threads if they are commented by the same user. The node attributes are constructed using the averaged word embedding vectors of the threads. We extract a subset of nodes from the original large network for the experiments.

Table 1: Statistics of datasets. r_1 denotes the ratio of the number of anomalies to the total number of nodes.

Datasets	Yelp	Amazon	PubMed	Reddit
# nodes	12, 671	14, 732	19, 717	22, 914
# edges	274, 842	201, 179	44, 326	135, 310
# features	8, 000	8, 000	500	602
# anomalies	765	705	1, 223	1, 306
r_1	6.04%	4.79%	6.20%	5.70%

Different from the *Yelp* and *Amazon* datasets, *PubMed* and *Reddit* do not contain ground-truth anomalies. Therefore, we employ two anomaly injection approaches [9, 44] to generate a combination of structural anomalies and contextual anomalies by modifying the graph topology and node attributes, respectively. To obtain structural anomalies, we adopt the method used by [9] to generate a set of cliques because a clique is often considered as a typical abnormal graph pattern where a group of nodes are much more closely connected to each other [43]. Concretely, to construct a clique, we randomly select c nodes (i.e., clique size) in the graph and then make these nodes fully linked to each other. By repeating this process K times (i.e., K cliques), we can obtain $K \times c$ structural anomalies. In our experiment, we choose the clique size c to be 15. Additionally, we build contextual anomalies following the method proposed by [44]. To be specific, we first randomly select a node v_i and then sample another 50 nodes from the graph. Among the 50 nodes, we choose the node v_j whose attributes (i.e., $\mathbf{x}_j$) have the largest Euclidean distance from $\mathbf{x}_i$. Then, we replace the attributes of node v_i with $\mathbf{x}_j$. Notably, the injected structural and contextual anomalies have the same quantity and the total number of injected anomalies is approximately 6% of the graph size.

Having obtained the ground-truth or injected node-level anomalies, we can now proceed to generate the labeled anomalous subgraphs (i.e., $\mathcal{B}_a$). We first randomly select k abnormal nodes from the graph as center nodes. Then we adopt random walk with restart (RWR) [47] to obtain a local subgraph. The length of random walk and the restart probability are set as 10 and 0.5, respectively, and the average size of the obtained subgraphs is around 8.67. Afterwards, the set of unlabeled subgraphs (i.e., $\mathcal{B}_n$) is constructed by applying RWR to the nodes sampled from the unlabeled set of nodes (i.e., $\mathcal{V}^u$). Particularly, for *Yelp* and *Amazon* datasets, in addition to the RWR-based strategy for constructing subgraphs, we also consider a subgraph to be a group of reviewers/customers (i.e., nodes) that post reviews on the same product/restaurant. We evaluate the performance of all supervised comparison methods on the two types of labeled subgraphs for *Yelp* and *Amazon* datasets. To denote the variants of subgraph type, we use "R" for RWR-generated subgraphs and "P" for subgraphs of connected reviewers on the same product.

Comparison Methods. We compare our proposed WEDGE framework with the following two groups of anomaly detection methods, including (1) *feature-based*: LOF [3], Autoencoder [68], DeepSAD [40] and MI-Net [22] where only the node attributes are used, and (2) *graph-based*: Radar [29], DOMINANT [8], SemiGNN [50], GDN [13], CARE-GNN [14], MI-GNN [48], CoLA [32], SL-GAD [67], and BWGNN [45] where both graph topological information and node attributes are considered. Note that for supervised methods designed for node-level labels, we consider all nodes in the labeled subgraphs as ground-truth anomalies. Details of the comparison methods are as follows.

- **LOF** [3] is a feature-based unsupervised approach which detects outliers based on the deviation of local density.
- **Autoencoder** [68] is a feature-based unsupervised deep autoencoder model which introduces an anomaly regularizing penalty based on L1 or L2 norms.
- **DeepSAD** [40] is a neural network-based approach for general semi-supervised anomaly detection. We use the node attributes as the input features in the experiment.
- **MI-Net** [22] is a deep multiple instance learning approach that incorporates attention mechanism for classification task. We use the node attributes as the training samples in our experiment.
- **Radar** [29] is an unsupervised method for anomaly detection on attributed network by characterizing the residuals of attribute and its consistency with network structure.
- **DOMINANT** [8] is a GCN-based autoencoder approach which computes anomaly scores according to the reconstruction errors from the perspectives of network structure and node attributes.
- **SemiGNN** [50] is a semi-supervised GNN model, which leverages the hierarchical attention mechanism to better correlate different neighbors and different views.
- **GDN** [13] is a recent GNN-based few-shot learning approach for node-level anomaly detection by enforcing large scores on anomalies with divergent behaviors.
- **CARE-GNN** [14] is a GNN-based anomaly detection model, which leverages reinforcement learning for selecting informative neighboring nodes according to a label-aware similarity measure.
- **MIL-GNN** [48] is a GNN-based multiple instance learning framework for graph classification. In the experiment, we consider each subgraph in the training set as an individual graph.
- **CoLA** [32] is a self-supervised GNN-based anomaly detection framework, which propose to sample effective instance pairs to capture the graph anomaly in a contrastive manner.
- **SL-GAD** [67] is an unsupervised graph anomaly approach which computes the anomaly scores from generative attribute reconstruction and multi-view contrastive learning modules.
- **BWGNN** [45] is a supervised method which proposes the spectral localized band-pass filters in GNN architectures tailored for graph anomaly detection.

Evaluation Metrics. To comprehensively evaluate the performance of different anomaly detection methods, in this work, we use metrics that are widely adopted by previous studies, including (1) AUC-ROC, (2) AUC-PR, and (3) Precision@K, detailed as follows.

- **AUC-ROC** is widely used in previous anomaly detection research [8, 29]. Specifically, Area under curve (AUC) depicts the

Table 2: Performance comparison results w.r.t. AUC-ROC and AUC-PR on four datasets.

Methods	Subgraph	Yelp		Amazon		PubMed		Reddit	
		AUC-ROC	AUC-PR	AUC-ROC	AUC-PR	AUC-ROC	AUC-PR	AUC-ROC	AUC-PR
LOF	R	0.351 ± 0.004	0.053 ± 0.003	0.467 ± 0.008	0.064 ± 0.003	0.542 ± 0.003	0.182 ± 0.007	0.497 ± 0.004	0.073 ± 0.002
Autoencoder	R	0.385 ± 0.009	0.042 ± 0.006	0.525 ± 0.012	0.103 ± 0.008	0.553 ± 0.010	0.242 ± 0.004	0.684 ± 0.006	0.341 ± 0.004
DeepSAD	R	0.471 ± 0.002	0.066 ± 0.007	0.458 ± 0.015	0.058 ± 0.007	0.511 ± 0.007	0.121 ± 0.004	0.518 ± 0.009	0.071 ± 0.005
MI-Net	R	0.569 ± 0.011	0.081 ± 0.006	0.587 ± 0.005	0.093 ± 0.013	0.624 ± 0.010	0.261 ± 0.011	0.694 ± 0.009	0.294 ± 0.013
	P	0.601 ± 0.008	0.075 ± 0.007	0.571 ± 0.013	0.087 ± 0.009	–	–	–	–
Radar	R	0.392 ± 0.005	0.041 ± 0.002	0.487 ± 0.009	0.069 ± 0.005	0.596 ± 0.007	0.211 ± 0.008	0.684 ± 0.003	0.251 ± 0.005
DOMINANT	R	0.575 ± 0.012	0.115 ± 0.006	0.652 ± 0.010	0.154 ± 0.002	0.641 ± 0.008	0.333 ± 0.010	0.710 ± 0.014	0.327 ± 0.007
SemiGNN	R	0.452 ± 0.008	0.042 ± 0.005	0.589 ± 0.014	0.105 ± 0.008	0.502 ± 0.011	0.067 ± 0.008	0.542 ± 0.007	0.115 ± 0.006
	P	0.467 ± 0.010	0.041 ± 0.003	0.601 ± 0.007	0.131 ± 0.011	–	–	–	–
GDN	R	0.613 ± 0.018	0.138 ± 0.012	0.684 ± 0.013	0.199 ± 0.016	0.698 ± 0.014	0.372 ± 0.009	0.725 ± 0.021	0.324 ± 0.014
	P	0.621 ± 0.011	0.140 ± 0.008	0.679 ± 0.008	0.211 ± 0.015	–	–	–	–
CARE-GNN	R	0.618 ± 0.017	0.119 ± 0.013	0.635 ± 0.011	0.192 ± 0.012	0.652 ± 0.011	0.356 ± 0.021	0.692 ± 0.010	0.319 ± 0.015
	P	0.631 ± 0.014	0.121 ± 0.008	0.661 ± 0.017	0.198 ± 0.011	–	–	–	–
MIL-GNN	R	0.535 ± 0.012	0.069 ± 0.013	0.498 ± 0.013	0.056 ± 0.004	0.591 ± 0.011	0.314 ± 0.015	0.634 ± 0.008	0.232 ± 0.016
	P	0.595 ± 0.021	0.082 ± 0.008	0.514 ± 0.014	0.064 ± 0.009	–	–	–	–
CoLA	R	0.613 ± 0.011	0.132 ± 0.007	0.651 ± 0.011	0.163 ± 0.013	0.632 ± 0.021	0.298 ± 0.013	0.668 ± 0.018	0.264 ± 0.011
SL-GAD	R	0.627 ± 0.015	0.145 ± 0.007	0.682 ± 0.014	0.159 ± 0.012	0.663 ± 0.021	0.337 ± 0.011	0.705 ± 0.017	0.335 ± 0.011
BWGNN	R	0.587 ± 0.012	0.135 ± 0.014	0.709 ± 0.017	0.208 ± 0.015	0.627 ± 0.025	0.311 ± 0.015	0.715 ± 0.018	0.317 ± 0.016
	P	0.607 ± 0.019	0.124 ± 0.07	0.679 ± 0.015	0.217 ± 0.013	–	–	–	–
WEDGE (ours)	R	0.721 ± 0.021	**0.181 ± 0.016**	**0.779 ± 0.016**	**0.331 ± 0.011**	**0.762 ± 0.013**	**0.412 ± 0.014**	**0.806 ± 0.025**	**0.401 ± 0.019**
	P	**0.735 ± 0.014**	0.169 ± 0.012	0.753 ± 0.017	0.318 ± 0.018	–	–	–	–

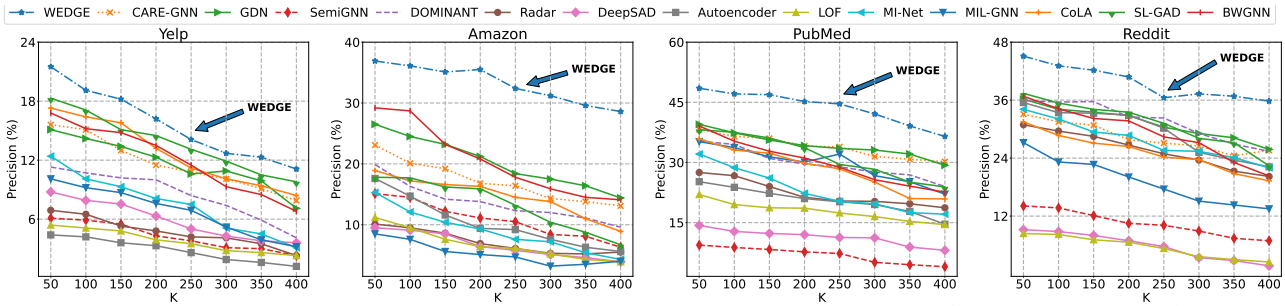

Figure 2: Performance comparison results w.r.t. Precision@K on four datasets. (Best viewed in color.)

probability that a randomly selected abnormal instance receives a higher score than a randomly chosen normal object.

- **AUC-PR** is the area under the curve of precision against recall at different thresholds, and it evaluates the performance on the positive class (i.e., abnormal objects).
- **Precision@K** is defined as the proportion of true anomalies in the top-ranked K objects. Specifically, we sort the anomaly scores from a detection algorithm in descending order.

Implementation Details. For pre-processing, we generate $k = 50$ anomalous subgraphs using the aforementioned strategy to obtain $\mathcal{B}_a$. For the proposed WEDGE framework, the *graph encoder* is a two-layer Graph Convolution Network (GCN) [25] with 512 dimension, followed by one hidden layer of size 256, as shown in Eq. (3). For the *subgraph abnormality predictor*, we choose the dimension of the attention vector (i.e., $\mathbf{p}$ in Eq. (5)) to be 128. The confidence margin (i.e., m in Eq. (9)) is set as 5 and the reference score (i.e., μ_r) is computed as the mean of 5,000 scores that are sampled from a Gaussian distribution, i.e., $\mathcal{N}(0, 1)$. We set the temperature $\tau = 0.1$ in Eq. (10), and the regularization parameter $\lambda = 0.4$ in Eq. (12).

For training, we sample $k = 50$ subgraphs to construct $\mathbf{B}_n$ and train the model with 1,000 epochs. For each epoch, we randomly select $N = 8$ subgraphs from $\mathcal{B}_a$ and $\mathcal{B}_n$, respectively, resulting in the batch size of 16. We use the Adam optimizer [24] with learning rate 0.01. The nodes are split into 40% for training, 20% for validation, and 40% for testing. For all comparison methods, we select the hyper-parameters with the best performance on the validation set and report the results on the test data. We report the average results after 10 runs of the training algorithm.

4.2 Effectiveness Results (RQ1)

We first evaluate the performance of the proposed framework WEDGE and the baseline methods in node-level anomaly detection. We present the evaluation results w.r.t. (1) AUC-ROC/AUC-PR in Table 2, and (2) Precision@K in Figure 2. Note that in Table 2, for supervised methods (i.e., MI-Net, Semi-GNN, GDN, CARE-GNN, MIL-GNN, BWGNN and WEDGE), we report the results for two types of subgraph-level labels on *Yelp* and *Amazon* datasets as introduced in Sec. 4.1, denoted by "R" and "P", respectively. We highlight the best performing method (i.e., WEDGE) in bold and underline the best comparison method, respectively. We have the following observations. First, in terms of AUC-ROC and AUC-PR, the proposed WEDGE outperforms all the comparison methods by a significant

Table 3: Performance of WEDGE w.r.t. AUC-ROC and AUC-PR at different levels of weak-supervision.

	Yelp		Amazon		PubMed		Reddit	
Length	AUC-ROC	AUC-PR	AUC-ROC	AUC-PR	AUC-ROC	AUC-PR	AUC-ROC	AUC-PR
5	0.738 ± 0.015	0.184 ± 0.018	0.785 ± 0.014	0.345 ± 0.018	0.775 ± 0.013	0.425 ± 0.010	0.811 ± 0.016	0.413 ± 0.015
10	0.721 ± 0.021	0.181 ± 0.016	0.779 ± 0.021	0.331 ± 0.011	0.762 ± 0.013	0.412 ± 0.014	0.806 ± 0.025	0.401 ± 0.019
15	0.717 ± 0.013	0.172 ± 0.008	0.752 ± 0.017	0.317 ± 0.015	0.748 ± 0.013	0.405 ± 0.011	0.798 ± 0.015	0.389 ± 0.007
20	0.695 ± 0.009	0.161 ± 0.011	0.747 ± 0.018	0.302 ± 0.014	0.740 ± 0.011	0.391 ± 0.012	0.785 ± 0.018	0.372 ± 0.014

margin. In addition, from the results w.r.t. Precision@K, WEDGE also achieves better performance in assigning higher anomaly scores to true anomalous nodes than other methods. Second, neither unsupervised methods (e.g., DOMINANT, Radar) or semi-supervised methods (e.g., DeepSAD, SemiGNN) deliver satisfactory results. The possible explanations are (1) unsupervised methods are incapable of leveraging the supervised knowledge of labeled anomalies; (2) for semi-supervised methods, DeepSAD cannot handle the topological information and SemiGNN requires a relatively large number of multi-view data with labels, which diminish the effectiveness of these methods. Third, existing supervised methods (e.g., GDN, BWGNN, CARE-GNN) can extract limited knowledge from the coarse-grained labels and hence have marginal improvement.

To corroborate the effectiveness of WEDGE in a weakly-supervised setting, we conduct experiments to evaluate the performance of WEDGE under different levels of weak-supervision. To be specific, we adopt the labeled subgraphs of various sizes by modifying the length of random walks in constructing subgraphs. We set the length of random walks l to be 5, 10, 15 and 20, respectively. Table 3 summarizes the results w.r.t. AUC-ROC/AUC-PR of WEDGE under different levels of weak supervision. We can observe that, in general, the model performance gradually decreases as the labeled subgraphs become larger in size. A possible explanation is that a larger anomalous subgraph may consist of more normal nodes, which has a negative impact on the quality of subgraph-level label (i.e., the supervision becomes even weaker). By comparing the results in Table 2 and Table 3, we can see that the proposed WEDGE is still able to considerably outperform baselines, which demonstrate the effectiveness of WEDGE under significantly weak supervision.

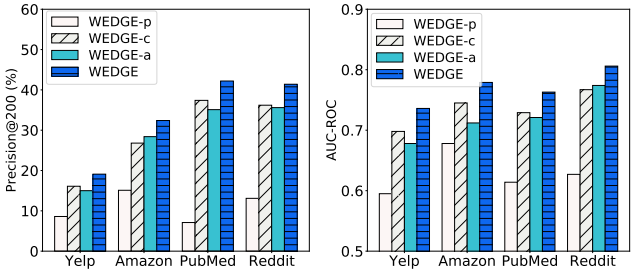

Figure 3: (a) Precision@200 of WEDGE and the variants; (b) AUR-ROC of WEDGE and the variants.

4.3 Ablation Study (RQ2)

In this section, we conduct an ablation study to inspect the contribution of each key component in WEDGE. We consider the following three variants of WEDGE, including (1) WEDGE-a that excludes the attention mechanism and utilizes the average pooling to compute the subgraph-level score/representation, (2) WEDGE-c that removes the contrastive objective during training, and (3) WEDGE-p that excludes both components. The results of performance w.r.t. Precision@200 and AUC-ROC are summarized in Figure 3a and Figure 3b, respectively. We have the following observations: (1) by comparing WEDGE-c and WEDGE-a with WEDGE-p, the attention-based method and the contrastive objective can separately improve the node anomaly detection by a remarkable margin. For example, the attention mechanism (i.e., WEDGE-c) can achieve 30% improvement in terms of Precision@200 on *PubMed* dataset over WEDGE-p. A possible reason is that the attention-based method can accurately extract the critical nodes inside an anomalous subgraph; and (2) the proposed WEDGE further benefits from the combination of the two components and consistently outperforms the variants. For instance, on *Yelp* dataset, WEDGE is 3.8% and 5.8% better in terms of AUR-ROC than WEDGE-c and WEDGE-a, respectively, which verifies the effectiveness of the key components in extracting critial knowledge and learning informative representations.

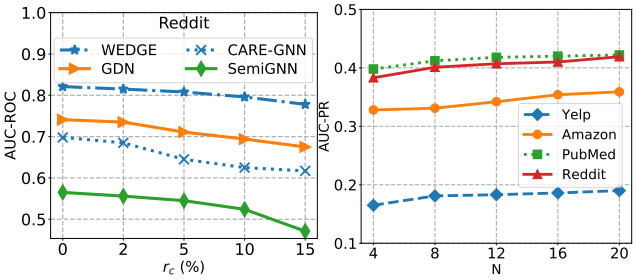

Figure 4: (a) Robustness study w.r.t. AUC-ROC with different contamination levels; (b) Sensitivity analysis w.r.t. AUC-PR with different batch size.

4.4 Robustness and Sensitivity Analysis (RQ3)

Moreover, we analyze the robustness and sensitivity of the proposed WEDGE framework. As mentioned in Sec. 3.1, in sampling subgraphs from the unlabeled node set $\mathcal{V}^u$, we treat all nodes in the sampled subgraph as normal, which could introduce contamination in the resulting set of subgraphs (i.e., $\mathcal{B}_n$). To investigate how robust our proposed WEDGE is w.r.t. different levels of contamination r_c (i.e., the proportion of anomalies in the unlabeled node set $\mathcal{V}^u$), we evaluate the performance of WEDGE, CARE-GNN, GDN and SemiGNN, and present the results w.r.t. AUC-ROC in Figure 4a. We can see that WEDGE is consistently robust to various levels of contamination and significantly outperforms other baselines.

The batch size N is an important hyper-parameter in contrastive learning [5]. We perform a sensitivity analysis by adjusting the batch size, and the sensitivity results are summarized in Figure 4b. We can observe that (1) in general, WEDGE benefits from larger batch size and can achieve a better performance in terms of AUC-PR, which can be attributed to that more negative instances are

included in a larger batch; and (2) the proposed WEDGE can still achieve a comparably good performance with a small training batch. For example, on *Reddit* dataset, AUC-PR only drops 0.036 in AUC-PR if we change the batch size from $N = 20$ to $N = 4$.

5 RELATED WORK

In this section, we review the related work in terms of (1) graph anomaly detection, (2) multiple instance learning, and (3) contrastive learning.

5.1 Graph Anomaly Detection

Graph anomaly detection approaches are specifically designed for graph structured data in the following two categories, (1) plain graphs with only topological information, and (2) attributed network with rich feature information of nodes/edges. For plain graphs, since the graph topology is the only available information, methods in this category aim to exploit the graph topological knowledge to identify anomalies [1]. In recent years, attributed networks have been widely adopted to model a variety of complex systems due to their strong capability for handling data heterogeneity [15, 51, 69]. Therefore, anomaly detection on attributed networks has drawn increasing research attention from the community [35, 41]. Among the proposed methods, ConOut [41] identifies the local context for each node and performs anomaly ranking within the local context. More recently, with the development of graph representation learning using neural networks, researchers propose to leverage graph neural networks (GNNs) for anomaly detection. DOMINANT [8] achieves remarkable performance over other shallow methods by building a deep autoencoder architecture with the graph convolutional networks. Semi-GNN [50] is a semi-supervised graph neural model which adopts hierarchical attention to model the multi-view graph for fraud detection. CARE-GNN [14] is a GNN-based fraud detector which improves the feature aggregation process by finding the optimal neighboring nodes. GAS [28] is a GCN-based large-scale anti-spam method for detecting spam advertisements. Zhao et al. propose an objective function to train GNNs for anomaly-detectable node representations [66]. In this work, we focus on detecting node anomalies with subgraph labels.

5.2 Multiple Instance Learning

Multiple instance learning (MIL) is one form of weakly-supervised learning where instances are organized in sets (bags) with which the coarse-grained labels are associated. In general, MIL research focuses on designing effective aggregation function to extract critical information from bags to infer unobserved bags [33] or instances [26]. For example, Xu et al. propose an averaging method to combine instance predictions using a logistic regression classifier [55]. Zhou et al. propose to use graph kernels to aggregate predictions by exploits the relations between instances [73]. More recently, aggregation function paramerized by deep neural networks has shown its superiority over shallow methods [18]. For instance, Ilse et al. propose an attention-based aggregation operator parameterized by neural networks which estimates the contribution of each instance to the bag prediction [22]. Tu et al. utilize GNNs to capture the correlation between nodes in order to generate graph prediction [48]. From the perspective of applications, MIL has been widely investigated, ranging from tumor image segmentation [56], object localization [6] to sentiment classification [2]. Different from the aforementioned MIL methods, our approach aims to capture the correlation between node and subgraph anomaly by exploiting the rich information from the attributed network.

5.3 Constrastive Learning

In recent years, contrastive learning has become a predominant topic in the field of self-supervised learning and it is extensively investigated in a variety of domains [5, 61]. The key idea of contrastive learning is enforcing the closeness between positive instances in the embedding space while segregating the samples in different categories. For instance, SimCLR aims to improve visual representation learning through contrasting images from various augmentation [5]. Tian et al. [46] study the contrastive learning in a multiview setting where they target at maximizing the mutual information between different views for capturing the scene semantics. CPC [37] is a universal unsupervised learning method which adopts a probabilistic contrastive loss to apprehend the most critical information for prediction. MoCo [21] leverages the idea of contrastive loss to construct large and consistent dynamic dictionary of instance-representation pairs, which improves the performance of various downstream tasks. More recently, contrastive learning is well-exploited to further enhance representation learning on graph structured data [20, 61, 75]. For example, Zhu et al. propose an unsupervised approach for graph learning by maximizing the agreement between the node embeddings in two graph views obtained by corruption [75]. Our work is related to contrastive learning in the way that we aim to improve the node representation learning for graph anomaly detection by contrasting subgraphs with different levels of abnormality.

6 CONCLUSION

In this paper, we study the challenging problem of weakly-supervised graph anomaly detection. We propose a novel graph neural network-based architecture, WEDGE, where the key innovation is to precisely quantify the intrinsic relationship between nodes and subgraph in terms of abnormalities and embeddings. The proposed WEDGE is able to enforce the large deviation from anomalous and normal nodes through extracting the critical nodes from weakly-supervised subgraph-level knowledge. To further improve the node representation learning, we design a contrastive objective, which aims to maximize the similarity between abnormal subgraphs and the disparity between subgraphs of different categories in the embedding space. We demonstrate the preeminence of WEDGE in node-level anomaly detection through extensive experimental evaluation.

ACKNOWLEDGEMENT

This work is supported by NSF (1947135, 2134079, 2316233, and 2324770), the NSF Program on Fairness in AI in collaboration with Amazon (1939725), DARPA (HR001121C0165), NIFA (2020-67021-32799), DHS (17STQAC00001-07-00), ARO (W911NF2110088), the C3.ai Digital Transformation Institute, and IBM-Illinois Discovery Accelerator Institute. The content of the information in this document does not necessarily reflect the position or the policy of the Government or Amazon, and no official endorsement should be inferred. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

REFERENCES

[1] Leman Akoglu, Hanghang Tong, and Danai Koutra. 2015. Graph based anomaly detection and description: a survey. *Data mining and knowledge discovery* (2015).

[2] Stefanos Angelidis and Mirella Lapata. 2018. Multiple instance learning networks for fine-grained sentiment analysis. *TACL* (2018).

[3] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. 2000. LOF: identifying density-based local outliers. In *SIGMOD*.

[4] Chen Chen, Hanghang Tong, Lei Xie, Lei Ying, and Qing He. 2016. FASCINATE: fast cross-layer dependency inference on multi-layered networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 765–774.

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *ICML*.

[6] Ramazan Gokberk Cinbis, Jakob Verbeek, and Cordelia Schmid. 2016. Weakly supervised object localization with multi-fold multiple instance learning. *TPAMI* (2016).

[7] Damon P Coppola. 2006. *Introduction to international disaster management*. Elsevier.

[8] Kaize Ding, Jundong Li, Rohit Bhanushali, and Huan Liu. 2019. Deep anomaly detection on attributed networks. In *SDM*.

[9] Kaize Ding, Jundong Li, and Huan Liu. 2019. Interactive anomaly detection on attributed networks. In *WSDM*.

[10] Kaize Ding, Yancheng Wang, Yingzhen Yang, and Huan Liu. 2023. Eliciting structural and semantic global knowledge in unsupervised graph contrastive learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 7378–7386.

[11] Kaize Ding, Zhe Xu, Hanghang Tong, and Huan Liu. 2022. Data augmentation for deep graph learning: A survey. *ACM SIGKDD Explorations Newsletter* 24, 2 (2022), 61–77.

[12] Kaize Ding, Chuxu Zhang, Jie Tang, Nitesh Chawla, and Huan Liu. 2022. Toward Graph Minimally-Supervised Learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4782–4783.

[13] Kaize Ding, Qinghai Zhou, Hanghang Tong, and Huan Liu. 2021. Few-shot Network Anomaly Detection via Cross-network Meta-learning. In *The Web Conference*.

[14] Yingtong Dou, Zhiwei Liu, Li Sun, Yutong Deng, Hao Peng, and Philip S Yu. 2020. Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. In *CIKM*.

[15] Boxin Du and Hanghang Tong. 2019. Mrmine: Multi-resolution multi-network embedding. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 479–488.

[16] Boxin Du, Si Zhang, Nan Cao, and Hanghang Tong. 2017. First: Fast interactive attributed subgraph matching. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 1447–1456.

[17] Boxin Du, Si Zhang, Yuchen Yan, and Hanghang Tong. 2021. New frontiers of multi-network mining: Recent developments and future trend. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 4038–4039.

[18] Ji Feng and Zhi-Hua Zhou. 2017. Deep MIML network. In *AAAI*.

[19] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *NeurIPS*.

[20] Kaveh Hassani and Amir Hosein Khasahmadi. 2020. Contrastive multi-view representation learning on graphs. In *ICML*.

[21] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *CVPR*.

[22] Maximilian Ilse, Jakub Tomczak, and Max Welling. 2018. Attention-based deep multiple instance learning. In *ICML*.

[23] Jian Kang, Qinghai Zhou, and Hanghang Tong. 2022. JuryGCN: quantifying jackknife uncertainty on graph convolutional networks. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 742–752.

[24] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[25] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.

[26] Dimitrios Kotzias, Misha Denil, Nando De Freitas, and Padhraic Smyth. 2015. From group to individual labels using deep features. In *KDD*.

[27] Hans-Peter Kriegel, Peer Kroger, Erich Schubert, and Arthur Zimek. 2011. Interpreting and unifying outlier scores. In *SDM*.

[28] Ao Li, Zhou Qin, Runshi Liu, Yiqun Yang, and Dong Li. 2019. Spam review detection with graph convolutional networks. In *CIKM*.

[29] Jundong Li, Harsh Dani, Xia Hu, and Huan Liu. 2017. Radar: Residual Analysis for Anomaly Detection in Attributed Networks.. In *IJCAI*.

[30] Yuncheng Li, Jianchao Yang, Yale Song, Liangliang Cao, Jiebo Luo, and Li-Jia Li. 2017. Learning from noisy labels with distillation. In *CVPR*.

[31] Yixin Liu, Kaize Ding, Huan Liu, and Shirui Pan. 2023. Good-d: On unsupervised graph out-of-distribution detection. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 339–347.

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

[33] Oded Maron and Aparna Lakshmi Ratan. 1998. Multiple-instance learning for natural scene classification.. In *ICML*.

[34] Julian John McAuley and Jure Leskovec. 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.

[35] Emmanuel Müller, Patricia Iglesias Sánchez, Yvonne Mülle, and Klemens Böhm. 2013. Ranking outlier nodes in subspaces of attributed graphs. In *ICDEW*.

[36] Eric WT Ngai, Yong Hu, Yiu Hing Wong, Yijun Chen, and Xin Sun. 2011. The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. *Decision support systems* 50, 3 (2011), 559–569.

[37] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[38] Guansong Pang, Chunhua Shen, and Anton van den Hengel. 2019. Deep anomaly detection with deviation networks. In *KDD*.

[39] Shebuti Rayana and Leman Akoglu. 2015. Collective opinion spam detection: Bridging review networks and metadata. In *KDD*.

[40] Lukas Ruff, Robert A Vandermeulen, Nico Görnitz, Alexander Binder, Emmanuel Müller, Klaus-Robert Müller, and Marius Kloft. 2019. Deep semi-supervised anomaly detection. *arXiv preprint arXiv:1906.02694* (2019).

[41] Patricia Iglesias Sánchez, Emmanuel Müller, Oretta Irmler, and Klemens Böhm. 2014. Local context selection for outlier ranking in graphs with multiple numeric node attributes. In *SSDBM*.

[42] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. 2008. Collective classification in network data. *AI magazine* (2008).

[43] David B Skillicorn. 2007. Detecting anomalies in graphs. In *ISI*.

[44] Xiuyao Song, Mingxi Wu, Christopher Jermaine, and Sanjay Ranka. 2007. Conditional anomaly detection. *TKDE* (2007).

[45] Jianheng Tang, Jiajin Li, Ziqi Gao, and Jia Li. 2022. Rethinking graph neural networks for anomaly detection. In *International Conference on Machine Learning*. PMLR, 21076–21089.

[46] Yonglong Tian, Dilip Krishnan, and Phillip Isola. 2019. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849* (2019).

[47] Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. 2006. Fast random walk with restart and its applications. In *ICDM*.

[48] Ming Tu, Jing Huang, Xiaodong He, and Bowen Zhou. 2019. Multiple instance learning with graph neural networks. *arXiv preprint arXiv:1906.04881* (2019).

[49] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. In *ICLR*.

[50] Daixin Wang, Jianbin Lin, Peng Cui, Quanhui Jia, Zhen Wang, Yanming Fang, Quan Yu, Jun Zhou, Shuang Yang, and Yuan Qi. 2019. A Semi-supervised Graph Attentive Network for Financial Fraud Detection. In *ICDM*.

[51] Haohui Wang, Yuzhen Mao, Jianhui Sun, Si Zhang, and Dawei Zhou. 2023. Dynamic Transfer Learning across Graphs. *arXiv preprint arXiv:2305.00664* (2023).

[52] Jianling Wang, Kaize Ding, Liangjie Hong, Huan Liu, and James Caverlee. 2020. Next-item recommendation with sequential hypergraphs. In *SIGIR*.

[53] Felix Wu, Tianyi Zhang, Amauri Holanda de Souza Jr, Christopher Fifty, Tao Yu, and Kilian Q Weinberger. 2019. Simplifying graph convolutional networks. *arXiv preprint arXiv:1902.07153* (2019).

[54] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. How powerful are graph neural networks?. In *ICLR*.

[55] Xin Xu and Eibe Frank. 2004. Logistic regression and boosting for labeled bags of instances. In *PAKDD*.

[56] Yan Xu, Jun-Yan Zhu, I Eric, Chao Chang, Maode Lai, and Zhuowen Tu. 2014. Weakly supervised histopathology cancer image segmentation and classification. *Medical image analysis* (2014).

[57] Zhe Xu, Yuzhong Chen, Qinghai Zhou, Yuhang Wu, Menghai Pan, Hao Yang, and Hanghang Tong. 2023. Node Classification Beyond Homophily: Towards a General Solution. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2862–2873.

[58] Yuchen Yan, Si Zhang, and Hanghang Tong. 2021. Bright: A bridging algorithm for network alignment. In *Proceedings of the Web Conference 2021*. 3907–3917.

[59] Yuchen Yan, Qinghai Zhou, Jinning Li, Tarek Abdelzaher, and Hanghang Tong. 2022. Dissecting cross-layer dependency inference on multi-layered interdependent networks. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 2341–2351.

[60] Jiaxuan You, Bowen Liu, Zhitao Ying, Vijay Pande, and Jure Leskovec. 2018. Graph convolutional policy network for goal-directed molecular graph generation. In *NeurIPS*.

[61] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph contrastive learning with augmentations. *(NeurIPS* (2020).

[62] Reza Zafarani, Mohammad Ali Abbasi, and Huan Liu. 2014. *Social media mining: an introduction*. Cambridge University Press.
[63] Shijie Zhang, Hongzhi Yin, Tong Chen, Quoc Viet Nguyen Hung, Zi Huang, and Lizhen Cui. 2020. Gcn-based user representation learning for unifying robust recommendation and fraudster detection. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 689–698.
[64] Si Zhang, Dawei Zhou, Mehmet Yigit Yildirim, Scott Alcorn, Jingrui He, Hasan Davulcu, and Hanghang Tong. 2017. Hidden: hierarchical dense subgraph detection with application to financial fraud detection. In *Proceedings of the 2017 SIAM International Conference on Data Mining*. SIAM, 570–578.
[65] Yin Zhang, Rong Jin, and Zhi-Hua Zhou. 2010. Understanding bag-of-words model: a statistical framework. *International Journal of Machine Learning and Cybernetics* (2010).
[66] Tong Zhao, Chuchen Deng, Kaifeng Yu, Tianwen Jiang, Daheng Wang, and Meng Jiang. 2020. Error-Bounded Graph Anomaly Loss for GNNs. In *CIKM*.
[67] Yu Zheng, Ming Jin, Yixin Liu, Lianhua Chi, Khoa T Phan, and Yi-Ping Phoebe Chen. 2021. Generative and contrastive self-supervised learning for graph anomaly detection. *IEEE Transactions on Knowledge and Data Engineering* (2021).
[68] Chong Zhou and Randy C Paffenroth. 2017. Anomaly detection with robust deep autoencoders. In *KDD*.
[69] Qinghai Zhou, Liangyue Li, Nan Cao, Lei Ying, and Hanghang Tong. 2019. ADMIRING: Adversarial multi-network mining. In *ICDM*.
[70] Qinghai Zhou, Liangyue Li, Nan Cao, Lei Ying, and Hanghang Tong. 2021. Adversarial Attacks on Multi-Network Mining: Problem Definition and Fast Solutions. *IEEE Transactions on Knowledge and Data Engineering* (2021).
[71] Qinghai Zhou, Liangyue Li, and Hanghang Tong. 2019. Towards Real Time Team Optimization. In *Big Data*.
[72] Qinghai Zhou, Liangyue Li, Xintao Wu, Nan Cao, Lei Ying, and Hanghang Tong. 2021. Attent: Active attributed network alignment. In *Proceedings of the Web Conference 2021*. 3896–3906.
[73] Zhi-Hua Zhou, Yu-Yin Sun, and Yu-Feng Li. 2009. Multi-instance learning by treating instances as non-iid samples. In *ICML*.
[74] Zhi-Hua Zhou, Min-Ling Zhang, Sheng-Jun Huang, and Yu-Feng Li. 2012. Multi-instance multi-label learning. *Artificial Intelligence* (2012).
[75] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2020. Deep graph contrastive representation learning. *arXiv preprint arXiv:2006.04131* (2020).