GCCAD: Graph Contrastive Coding for Anomaly Detection

Bo Chen, Jing Zhang*, Xiaokang Zhang, Yuxiao Dong, Jian Song, Peng Zhang, Kaibo Xu, Evgeny Kharlamov, and Jie Tang*, *Fellow, IEEE*

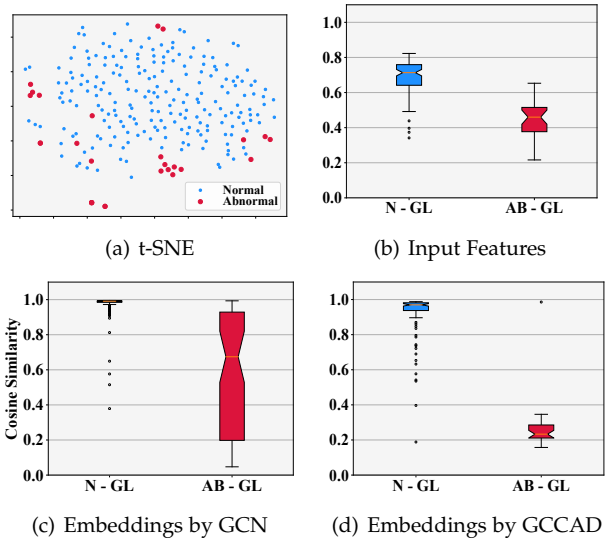
Abstract—Graph-based anomaly detection has been widely used for detecting malicious activities in real-world applications. Existing attempts to address this problem have thus far focused on structural feature engineering or learning in the binary classification regime. In this work, we propose to leverage graph contrastive coding and present the supervised GCCAD model for contrasting abnormal nodes with normal ones in terms of their distances to the global context (e.g., the average of all nodes). To handle scenarios with scarce labels, we further enable GCCAD as a self-supervised framework by designing a graph corrupting strategy for generating synthetic node labels. To achieve the contrastive objective, we design a graph neural network encoder that can infer and further remove suspicious links during message passing, as well as learn the global context of the input graph. We conduct extensive experiments on four public datasets, demonstrating that 1) GCCAD significantly and consistently outperforms various advanced baselines and 2) its self-supervised version without fine-tuning can achieve comparable performance with its fully supervised version.

Index Terms—Graph Neural Network, Anomaly Detection, Contrastive Learning

1 Introduction

Anomaly detection has profound impacts on preventing malicious activities in various applications, such as the detection of online review spams [26], financial frauds [27], [46], fake users [11], and misinformation [8], [39]. The most promising developments have been the utilization of graph structures in machine learning models for distinguishing the anomalies from the normal nodes, as graphs can be used for naturally modeling the structural dependencies underlying the data [2], [10], [24], [31].

Recently, the advances of graph neural networks (GNNs) [13], [21], [43] have inspired and empowered various attempts to adopt GNNs for detecting anomalies [10], [29], [31], [46]. The main idea of GNN-based anomaly detection is to leverage the power of GNNs to learn expressive node representations with the goal of distinguishing abnormal nodes from normal ones in the embedding space. Most of the GNN models are based on the inductive bias that two neighboring nodes tend to have the same labels. However, the suspicious links between the abnormal and normal nodes violate the above assumption, making GNNs produce confusing node embeddings.



(a) t-SNE (b) Input Features (c) Embeddings by GCN (d) Embeddings by GCCAD

Fig. 1. A real example of detecting the papers (red) that don't belong to "Jun Lu". (a) The t-SNE projection of the graph of all his papers, wherein nodes are papers and if two papers share the same coauthors, affiliations, or venues, an edge links them; (b) The similarities between normal nodes and the global context (blue) vs. that between abnormal nodes and the global context (red) by using input features (N, AB, GL are abbreviated for Normal, ABnormal nodes, and GLobal context, respectively); (c) The similarities between embeddings generated by GCN. (d) The similarities between embeddings generated by GCCAD.

To further understand the behavior of anomalies, we take the author "Jun Lu", a professor from Yale University, from Google Scholar's published author profile data as a case study by building a graph with papers assigned to

- *Bo Chen and Peng Zhang are with Department of Computer Science and Technology, Tsinghua University, Beijing, China, 100084. E-mail: {cb21, zhangp18}@mails.tsinghua.edu.cn,*
- *Jing Zhang and Xiaokang Zhang are with Information School, Renmin University of China, Beijing, China. E-mail:{zhang-jing, zhang2718}@ruc.edu.cn*
- *Yuxiao Dong is with Facebook AI, Seattle, USA. Email: ericdongyx@gmail.com*
- *Jian Song is with Zhipu.AI, Beijing, China. Email: sxusjj@gmail.com.*
- *Kaibo Xu is with Mininglamp Technology, Beijing, China. Email: xukaibo@mininglamp.com.*
- *Evgeny Kharlamov is with Bosch Center for Artificial Intelligence, Renningen, Germany. Email: Evgeny.Kharlamov@de.bosch.com and University of Oslo, Norway. Email: Evgeny.Kharlamov@ifi.uio.no.*
- *Jie Tang is with Department of Computer Science and Technology, Tsinghua University, and Tsinghua National Laboratory for Information Science and Technology (TNList), Beijing, China, 100084. E-mail: jietang@tsinghua.edu.cn,*

**Corresponding author*

him as nodes, and connect two papers with an edge if they share the same coauthors, affiliations, or venues. The goal here is to detect the wrongly assigned papers (anomalies), i.e., those are not authored by him. This anomaly detection problem can be motivated by a news[1] reported in 2012 that another researcher named "Jun Lu" from Beijing University of Chemical Technology cheated the awards using the papers of "Jun Lu" from Yale. This event caused by the wrongly assigned papers of the same author name implies the importance of anomaly detection. Thus, we illustrate how the wrongly assigned papers can be distinguished from the right ones in Figure 1. Figure 1(a) shows the t-SNE[2] projection of each node's input features—the BERT embedding [9] of its title—with blue as normal nodes and red as abnormal ones. We observe both abnormal and normal nodes are distributed diversely with abnormal ones being relatively more diverse. Intuitively, we quantify this observation by computing the similarity between each node and the global context—the average of all node features, which is shown in Figure 1(b). It suggests that though having slight overlaps, the two similarity distributions can be clearly distinguished. Inspired by these observations, we explore whether there is a straightforward way to capture them for distinguishing abnormal nodes from normal ones.

Present Work. In light of the recent progress in contrastive learning [14], [53], we propose to contrast each node with the global context of the input graph. The underlying assumption is that abnormal nodes tend to be more distant from the majority of the nodes, namely the graph context, than normal ones. We name the model as GCCAD, a Graph Contrastive Coding for Anomaly Detection. Specifically, we design the context-aware graph contrastive loss function in a supervised manner, i.e., labeled normal and abnormal nodes are treated as positive and negative keys respectively (Cf. Section 2.2), differing it from most existing studies that use graph contrastive learning in a self-supervised pre-training setup [35], [44], [58]. Figures 1(d) plots the similarity distributions of embeddings generated by GCCAD compared with that by GCN [21] shown in Figures 1(c), demonstrating GCCAD's striking capacity of separating abnormal nodes from normal ones when considering their distances to the global context of the graph.

In addition to this main (supervised) GCCAD model, we also extend it as an unsupervised pre-training model GCCAD-pre for handling cases with scarce labels. Straightforwardly, we need to synthesize node labels that can be directly used to replace the ground-truth labels in the (supervised) contrastive loss function of GCCAD. To generate synthetic abnormal nodes, we design a strategy to corrupt each part of the original graph by injecting the nodes outside this part (Cf. Section 2.3).

To achieve the contrastive objective, we propose a context-aware GNN encoder with three modules: *edge update*, *node update*, and *graph update*. First, *edge update* is used to estimate the suspicious likelihood of each link and then update the adjacency matrix of the graph by removing the most suspicious links. Then, *node update* is to update the node embeddings by message passing on the updated adjacency matrix. Finally, *graph update* is designed to update the global context iteratively.

We verify the proposed model by two genres of anomaly detection tasks, i.e., detecting the wrongly assigned papers in researchers' profiles on two academic datasets — AMiner and MAS, and detecting users who give fraudulent ratings on two business websites — Alpha and Yelp. The academic datasets have multiple author profiles with each of them can be viewed as a graph, dubbed as the multi-graph settings, and the business datasets only have one large graph dubbed as the single-graph settings (Cf. Section 2.3). Experiment results show that 1) GCCAD yields substantial improvements on the multi-graph datasets, while presenting subtle but consistent performance gain on the single-graph datasets compared with the state-of-the-art baselines and 2) the unsupervised GCCAD-pre is comparable with the fully-supervised GCCAD. With further fine-tuning, GCCAD-pre can outperform GCCAD on most of the datasets. In general, GCCAD-pre achieves consistently better performance than all the comparison graph pre-training methods.

The main contributions are summarized as follows:

- We propose the idea of using graph contrastive coding for anomaly detection and present GCCAD by designing context-aware graph contrastive objective.
- We design an effective strategy to generate synthetic labels for extending GCCAD as an unsupervised framework GCCAD-pre.
- We devise a context-aware GNN encoder through injecting context information to generate both node and context representations.
- We conduct experiments, demonstrating the substantial improvements brought by GCCAD and GCCAD-pre.

2 GCCAD: Graph Contrastive Coding for Anomaly Detection

We present GCCAD, the graph contrastive coding model, in this section. We first introduce the problem definition of anomaly detection (Section 2.1), and then conduct the preliminary observations to verify the motivation of GCCAD (Section 2.1). After that, we propose the learning objective with theoretical guarantees (Section 2.2), and further extend the supervised objective function to the unsupervised setting (Section 2.3). Finally, we introduce the proposed context-aware GNN encoder for GCCAD and GCCAD-pre (Section 2.4).

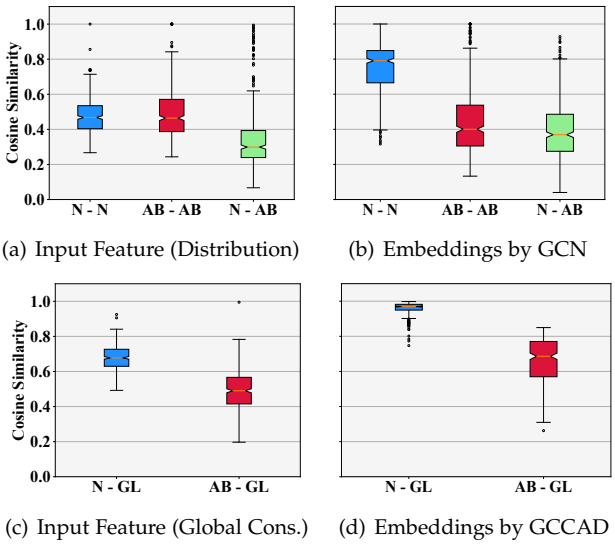
2.1 The Studied Problem

We formalize the problem of graph-based anomaly detection. We define a graph as $G = (V, X, A, Y)$, where V is the set of N nodes, $A \in \mathbb{R}^{N \times N}$ denotes the adjacency matrix, and $X \in \mathbb{R}^{N \times d}$ is the corresponding feature vectors with $\boldsymbol{x}_i \in \mathbb{R}^d$ representing the d-dimensional feature vector of node v_i. Without loss of generality, we consider G into an undirected and single-relational graph, i.e., $A_{ij} > 0$ if there exists an edge between v_i and v_j and $A_{ij} = 0$ otherwise.

Problem 1. **Graph-based Anomaly Detection.** Given a labeled graph $G = (V, X, A, Y)$, Y is the set of node labels with $y_i \in Y$ equals to 1 if v_i is abnormal and 0

1. https://www.universityworldnews.com/post.php?story=20120807160325397

2. https://lvdmaaten.github.io/tsne/



(a) Input Feature (Distribution) (b) Embeddings by GCN

(c) Input Feature (Global Cons.) (d) Embeddings by GCCAD

Fig. 2. (a) The similarities between normal and normal nodes (blue), abnormal and abnormal nodes (red), and normal and abnormal nodes (green) by the BERT-initialized input features; (b) The similarities between embeddings generated by GCN. (c) The similarities between normal nodes and the global context (blue) vs. those between abnormal nodes and the global context (red) by the input features; (d) The similarities between embeddings generated by GCCAD.

otherwise. The goal is to learn a function $g : \mathbb{R}^d \to \{0, 1\}$ to determine whether a given node is abnormal (1) or normal (0).

To resolve this problem, most existing GNN-based models directly instantiate g as a binary classifier [10], [31]. We conduct the following preliminary observations to verify the motivation of the proposed model.

Preliminary Observations. To further verify the motivation of contrasting a node with the global context in Figure 1, we additionally extract 10,000 authors owning more than 500,000 papers from AMiner[3], a free online academic search and mining system. Likewise, the goal here is to detect the wrongly assigned papers (anomalies). For each author, the wrongly assigned papers (anomalies) are labeled by professional annotators or the author themselves. For each paper, we obtain its BERT embedding [9] by its title. Then we calculate the cosine similarity between a pair of papers and average the pairwise cosine similarities of three groups, i.e., Normal and Normal (N-N), ABnormal and ABnormal (AB-AB), and Normal and ABnormal (N-AB). Figure 2(a) shows the following phenomenon.

- **Intra-Diversity:** The similarities in both N-N and AB-AB are extremely diverse (being scattered within [0.2,0.8]), which is in consistent with the case shown in Figure 1(a);
- **Inter-Diversity:** The similarities of N-AB are also diverse, and more than 20% abnormal nodes are similar to the normal nodes (y $>$ 0.5).

We conjecture that this intra-/inter- diversity will impact the performance of distinguishing the anomalies from the normal ones by the traditional classifier. To verify the conjecture, we investigate the same similarities as based on the embeddings generated by GCNs [21] with a binary classification loss function. The results shown in Figure 2(b) demonstrates that although the similarities in N-N increase from [0.2, 0.8] to [0.4, 1.0], the intra- and inter- diversity issues are still severe.

Previous efforts [17], [24], [31] disclose the behavior patterns of anomalies are different from that of normal nodes, based on which they characterize the inductive bias to detect anomalies. However, most of them only focus on modifying the message passing process of GNNs to reduce the propagated noises, while ignoring the limited capability of the binary classification loss function when the data distribution is diverse (Figure 2(a), 2(b)).

Inspired by the case observed in Figure 1(b), we compute the similarity between each node and the global context—the average of all the node features[4], and show the average similarities in N-GL and AB-GL in Figure 2(c). We can see that compared with Figure 2(a), although still having overlaps, the two similarity distributions can be distinguished much more clearly. Furthermore, we plot the similarities based on the embeddings generated by the proposed GCCAD in Figure 2(d). Compared with Figure 2(c), the resultant embeddings of the normal and abnormal ones by GCCAD can be futher distinguished.

In summary, we empirically verify the motivation of the context-aware contrastive learning method, that is, instead of directly capturing the absolute difference between normal and abnormal nodes, contrasting the relative distances between the nodes and global context may be a better way to address the diversity of absolute node distributions.

2.2 The GCCAD Model

The basic idea of GCCAD is to determine a node's abnormality by contrasting it with the global context (e.g., the average of all nodes) of the entire graph. This is motivated by the discovery that there exists a significant difference of distance to the global context between the normal and the abnormal nodes, that is, a node is more abnormal if it deviates farther away from the majority of nodes in the feature space.

In view of this, we distinguish abnormal and normal nodes in the embedding space by leveraging the graph contrastive coding (GCCAD). Specifically, given a graph G, we first create a GNN encoder f_{GNN} that can output an embedding $\boldsymbol{h}_i$ for each node v_i and also an embedding $\boldsymbol{q}$ for the entire graph (global context), i.e., $(H, \boldsymbol{q}) = f_{\text{GNN}}(X, A, W)$ with $H = \{\boldsymbol{h}_i\}_{i=1}^N$ and W as the trainable parameters of f_{GNN}. We take the graph embedding $\boldsymbol{q}$ as the query, a normal node's embedding as the positive key that matches with $\boldsymbol{q}$, and the embeddings of all the abnormal nodes as the negative keys that don't match with $\boldsymbol{q}$. For implementation, we use infoNCE [34] in a *supervised* manner as the concrete loss function such that:

$$\mathcal{L}_{\text{con}} = \mathop{\mathbb{E}}_{\substack{i:y_i=0 \\ j:y_j=1}} \left[-\log \frac{\exp\left(\mathbf{q}^\top \boldsymbol{h}_i/\tau\right)}{\sum_j \exp\left(\boldsymbol{q}^\top \boldsymbol{h}_j/\tau\right) + \exp\left(\boldsymbol{q}^\top \boldsymbol{h}_i/\tau\right)} \right] \quad (1)$$

3. https://www.aminer.org/

4. The global context is later estimated by the memory-based context updater.

where τ is the temperature hyperparameter. The objective function is to enforce maximizing the consistency between the positive pairs (normal node, global context) compared with negative pairs (abnormal node, global context).

Why does GCCAD work? We theoretically prove the effectiveness of the proposed GCCAD compared with the original cross-entropy loss for classfication.

Theorem 1. Let X^{+}[5] denote the random variable of normal node and $p_{\text{n}}(\cdot)$ denote its marginal distribution, thus $X^{+} \sim p_{\text{n}}(x^{+})$. Likewise, we denote the abnormal node by X^{-} and its marginal distribution by $p_{\text{ab}}(\cdot)$, $X^{-} \sim p_{\text{ab}}(x^{-})$. Assume p_{n} and p_{ab} are mutually independent. Then we have: *minimizing the contrastive loss in Eq.* (1) *forms the lower bound of 1) Kullback-Leibler (KL) divergence of the two data distributions $p_n(x^{+})$ and $p_{ab}(x^{-})$ and 2) the entropy of $p_{ab}(x^{-})$*. Formally,

$$\min \mathcal{L}_{\text{con}} \triangleq \max \left[D_{KL}(X^{-}||X^{+}) + 2H(X^{-}) \right]. \tag{2}$$

Considering the first part of the KL divergence in Eq (2), admittedly, maximizing the KL divergence D_{KL} between two distributions equals to maximizing the cross-entropy H[6] between them if one of the distribution keeps unchanged, i.e.,

$$D_{KL}(X^{-}||X^{+}) \triangleq \begin{cases} H(X^{-}, X^{+}) & p_{\text{ab}}(x^{-}) \text{ is static,} \\ D_{KL}(X^{-}||X^{+}) & \text{otherwise.} \end{cases} \tag{3}$$

The abnormal distribution $p_{\text{ab}}(\cdot)$ can be dynamically changed or keep statically under different scenarios. For example, when we detect the wrongly assigned papers for different authors, the feature distributions of the wrong papers are dynamically changed according to different authors. On the contrary, when we detect the fraudulent users in a social network, the abnormal distribution is usually unchanged. Cross-entropy only considers the scenario when the abnormal distribution is static, but ignores the dynamically changed scenario, while the KL divergence includes both the scenarios, which is more resilient than cross-entropy.

Considering the second part of the entropy in Eq (2), maximizing the entropy of the abnormal node distribution $H(X^{-})$ is in concert with some data augmentation methods, which augment and preserve the information of minority classes for solving the problem of class imbalance [50], [56].

In view of the two parts in Eq (2), the contrastive loss $\mathcal{L}_{\text{con}}$ is theoretically and analytically robust and superior than the cross-entropy loss.

Proof of Eq. (2). Let $\boldsymbol{h}^{+} = f_{\text{GNN}}(x^{+}, \cdot, W)$, where $\boldsymbol{h}_i^{+} \in \mathbb{R}^d$, then we have the normal node embedding matrix $H^{+} = \{\boldsymbol{h}_i^{+}\}_{i=1}^{n}$, where n is the number of normal nodes. Likewise, we define $\boldsymbol{h}^{-} = f_{\text{GNN}}(x^{-}, \cdot, W)$ and the abnormal node embedding matrix $H^{-} = \{\boldsymbol{h}_j^{-}\}_{j=1}^{m}$ with the node number m. Note that, $n \gg m$. Let $\mathcal{R}(\cdot)$ be a deterministic readout function on graphs. Without loss of generality, we assume $\mathcal{R}(\cdot) = \text{MEAN}(\cdot)$. Thus $\boldsymbol{q} = \mathcal{R}(H) = \frac{1}{N}(\sum_{i=1}^{n} \boldsymbol{h}_i^{+} + \sum_{j=1}^{m} \boldsymbol{h}_j^{-})$. From Eq. (1), we can derive

$$\begin{aligned} \mathcal{L}_{\text{con}} &= \underbrace{\mathop{\mathbb{E}}_{x^{+}\sim p_{\text{n}}} \left[-\boldsymbol{q}^T \boldsymbol{h}^{+}/\tau \right]}_{\text{alignment}} \\ &+ \underbrace{\mathop{\mathbb{E}}_{\substack{x^{+}\sim p_{\text{n}}, \\ \{x_j^{-}\}_{j=1}^{m}\sim p_{\text{ab}}}} \left[\log \left(e^{\boldsymbol{q}^T \boldsymbol{h}^{+}/\tau} + \sum_j e^{\boldsymbol{q}^T \boldsymbol{h}_j^{-}/\tau} \right) \right]}_{\text{uniformity}}, \end{aligned} \tag{4}$$

where the "alignment" term pulls the distances between the normal nodes and the global context closer, and the "uniformity" term pushes the distances between the abnormal nodes and the global context away [48].

Since $\boldsymbol{q} = \mathcal{R}(H) = \frac{1}{N}(\sum_{i=1}^{n} \boldsymbol{h}_i^{+} + \sum_{j=1}^{m} \boldsymbol{h}_j^{-})$, we get

$$\boldsymbol{q}^T \boldsymbol{h}^{+} = \frac{1}{N} \left[\sum_{i=1}^{n} (\boldsymbol{h}_i^{+})^T \boldsymbol{h}^{+} + \sum_{j=1}^{m} (\boldsymbol{h}_j^{-})^T \boldsymbol{h}^{+} \right]. \tag{5}$$

Note that $n \gg m$ and learned by Inter-/Intra- Diversity observations, the similarity scores between normal nodes is predominant and usually large, that is, the term $\boldsymbol{q}^T \boldsymbol{h}^{+}$ is naturally large, thus the main challenge lies in optimizing the "uniformity". Asymptotically, suppose the normal node pairs are perfectly aligned, i.e., $\boldsymbol{q}^T \boldsymbol{h}^{+} = 1$, then minimizing Eq. (4) is equivalent to optimizing the second term, i.e.

$$\begin{aligned} \mathcal{L}_{\text{con}} &\triangleq \mathop{\mathbb{E}}_{\substack{x^{+}\sim p_{\text{n}}, \\ \{x_j^{-}\}_{j=1}^{m}\sim p_{\text{ab}}}} \left[\log \left(e^{1/\tau} + \sum_j e^{\boldsymbol{q}^T \boldsymbol{h}_j^{-}/\tau} \right) \right] \\ &\geq \mathop{\mathbb{E}}_{\substack{x^{+}\sim p_{\text{n}}, \\ \{x_j^{-}\}_{j=1}^{m}\sim p_{\text{ab}}}} \left[\log \left(m(\frac{\sum_j e^{\boldsymbol{q}^T \boldsymbol{h}_j^{-}/\tau}}{m}) \right) \right] \\ &\geq \mathop{\mathbb{E}}_{\substack{x^{+}\sim p_{\text{n}}, \\ \{x_j^{-}\}_{j=1}^{m}\sim p_{\text{ab}}}} \left[\frac{\sum_j \log e^{\boldsymbol{q}^T \boldsymbol{h}_j^{-}/\tau}}{m} + \log m \right] \\ &\geq \mathop{\mathbb{E}}_{\substack{x^{+}\sim p_{\text{n}}, \\ \{x_j^{-}\}_{j=1}^{m}\sim p_{\text{ab}}}} \left[\frac{1}{m} \sum_j \left(\boldsymbol{q}^T \boldsymbol{h}_j^{-}/\tau \right) \right], \\ &\triangleq \mathop{\mathbb{E}}_{\substack{x^{+}\sim p_{\text{n}}, \\ \{x_j^{-}\}_{j=1}^{m}\sim p_{\text{ab}}}} \frac{1}{N} \left[\frac{1}{m} \sum_j \left(\sum_{i=1}^{n} (\boldsymbol{h}_i^{+})^T \boldsymbol{h}_j^{-} + \sum_{k=1}^{m} (\boldsymbol{h}_k^{-})^T \boldsymbol{h}_j^{-} \right) / \tau \right] \end{aligned} \tag{6}$$

where the second inequality follows the Jensen Inequality based on the concavity of the log function, namely $\log(\mathbb{E}[x]) \geq \mathbb{E}[\log(x)]$.

As p_{n} and p_{ab} are mutually independent and the data samples from either p_{n} or p_{ab} follow i.i.d. assumptions, minimizing the last equation of Eq. (6) is equivalent to both minimizing the sum of similarities between normal and abnormal node embeddings and minimizing the sum of the similarities between abnormal and abnormal node embeddings, i.e.

5. We denote random variables using upper-case letters (e.g. X^{+}, X^{-}), and their realizations by the corresponding lower-case letter (e.g. x^{+}, x^{-}).

6. As [6] claims, the standard cross entropy loss implicitly maximizes the inter-class distance, which can be denoted as $H(X^{-}, X^{+})$ here.

$$\min \text{Eq. (6)} \triangleq \min\left[\frac{1}{m}\sum_j\left(\sum_{i=1}^{n}(\boldsymbol{h}_i^+)^T\boldsymbol{h}_j^- + \sum_{k=1}^{m}(\boldsymbol{h}_k^-)^T\boldsymbol{h}_j^-\right)/\tau\right] \quad (7)$$

$$\triangleq \frac{1}{m}\sum_j\left[\min\left(\sum_{i=1}^{n}(\boldsymbol{h}_i^+)^T\boldsymbol{h}_j^-\right)/\tau + \min\left(\sum_{k=1}^{m}(\boldsymbol{h}_k^-)^T\boldsymbol{h}_j^-\right)/\tau\right]$$

$$\triangleq \frac{1}{m}\sum_j\left[\min\left(\log\frac{1}{n}\sum_{i=1}^{n}e^{(\boldsymbol{h}_i^+)^T\boldsymbol{h}_j^-/\tau} + \log n\right) + \min\left(\log\frac{1}{m}\sum_{k=1}^{m}e^{(\boldsymbol{h}_k^-)^T\boldsymbol{h}_j^-/\tau} + \log m\right)\right],$$

where the expectation symbol in the first equality is omit for brevity, and the last equality is obtained by first applying exponent to the each of similarity score, and then re-scaling the expectation of the similarity scores in the logarithm form. Notably, the last equation holds true under the minimization optimization circumstance.

Inspired by the entropy estimation operation in [48], the last equality can be also viewed as a resubstitution entropy estimator of $\boldsymbol{h}^{+/-}$ [1] via a von Mises-Fisher (vMF) kernel density estimation (KDE), formally,

$$\min \text{Eq. (7)} \triangleq \min\left[\frac{1}{m}\sum_j \log\hat{p}_{vMF}^{+}(\boldsymbol{h}^+)\right] + \min\left[\frac{1}{m}\sum_j \log\hat{p}_{vMF}^{-}(\boldsymbol{h}^-)\right] + \log Z_{vMF} \quad (8)$$

$$\triangleq \min\left[-\hat{H}(\boldsymbol{h}^-,\boldsymbol{h}^+) - \hat{H}(\boldsymbol{h}^-) + \log Z_{vMF}\right]$$

$$\triangleq \min\left[-\hat{D}_{KL}(\boldsymbol{h}^-||\boldsymbol{h}^+) - 2\hat{H}(\boldsymbol{h}^-) + \log Z_{vMF}\right]$$

where $\hat{p}_{vMF}^{+/-}$ is the KDE based on samples $H^{+/-}$ using a vMF kernel τ^{-1}, $\log Z_{vMF}$ is the normalization constant for vMF distribution, $\hat{H}$ and $\hat{D}_{KL}$ denote the resubstitution entropy estimator.

Admittedly, KL divergence and entropy are invariant under reparametrization of the variables [23], namely if $\boldsymbol{h}^+ = f_{\text{GNN}}(x^+,\cdot,W)$ and $\boldsymbol{h}^- = f_{\text{GNN}}(x^-,\cdot,W)$ are homeomorphisms (i.e. smooth invertible maps), we have $\hat{D}_{KL}(x^-||x^+) = \hat{D}_{KL}(\boldsymbol{h}^-||\boldsymbol{h}^+)$ and $\hat{H}(x^{+/-}) = \hat{H}(\boldsymbol{h}^{+/-})$. Thus,

$$\min \mathcal{L}_{\text{con}} \triangleq \max\left[D_{KL}(X^-||X^+) + 2H(X^-)\right]. \quad (9)$$

The results reveal that minimizing the contrastive loss in Eq. (1) is equivalent to maximizing the KL divergence between the two data distributions, $p_{\text{n}}(x^+)$ and $p_{\text{ab}}(x^-)$ and the entropy of $p_{\text{ab}}(x^-)$.

Connection to Existing Graph CL Frameworks. GCCAD, that performs contrastive learning in a supervised manner, differs from existing graph contrastive learning frameworks for GNN pre-training, such as GCC [35], DGI [44], GraphCL [58], because most of them are self-supervised and the contrastive instances should be constructed from the unlabeled data. For example, in GCC [35], given the embedding of a randomly sampled ego-network of a node as the query, the positive key is the embedding of another sampled ego-network of the same node, and the negative keys are those sampled from other nodes. In DGI [44], given the entire graph embedding as the query, the positive key is the embedding of a node in it, and the negative keys are those of the same node after permutating the graph. GraphCL [58] further contrasts between different graph augmentations. Given the embedding of a graph as the query, the positive key is the embedding of the graph augmented from the same graph, while the negative keys are those from different graphs. In contrast, in our objective for anomaly detection, given the graph embedding as the query, the positive and the negative keys are constrained to the normal and abnormal nodes respectively. In light of this, GCCAD is a kind of *supervised* contrastive learning [20].

2.3 GCCAD-pre for Unsupervised Settings

Sufficient labels of anomalies are often arduously expensive to obtain, which is the bottleneck of most anomaly detection scenarios. Therefore, we further extend GCCAD as an unsupervised pre-training model (GCCAD-pre) to handle real-world applications with scarce labels.

We present a pre-training strategy for tackling the label scarce problem in anomaly detection. Inspired by the idea of context-aware contrastive objective, we propose to construct pseudo anomalies via corrupting the original graph. Considering one small part of the original graph, we inject nodes outside this part into it. The underlying assumption is that as nodes in different parts of the graph follow different distributions [35], [44], they can serve as the pseudo anomalies to the context of the small part of the original graph. Formally, a corrupt graph is defined as follows:

Definition 1. **Corrupt Graph.** Given a graph $G = (V, X, A)$, we break it into M parts $\{G_i\}_{i=1}^{M}$. For each part G_i, a corrupt graph $\tilde{G}_i$ is created by injecting a set of nodes $\bar{V}_i$ from the parts except G_i, i.e., $\bar{V}_i = \{v_j \in G \setminus G_i\}$, thus the corrupt nodes $\tilde{V}_i$ are the union of V_i and $\bar{V}_i$, i.e., $\tilde{V}_i = V_i \cup \bar{V}_i$. The corrupt adjacency matrix $\tilde{A}_i$ of $\tilde{G}_i$ is obtained by slicing A using the indexes in $\tilde{V}_i$.

Figure 3 illustrates a toy example of creating corrupt graphs. There are different ways for breaking the input graph into multiple parts w.r.t various graph distributions. Generally, these different ways can be summarized into two categories:

- **Single-Graph:** If the original graph is a single graph, we partition it into multiple parts using clustering algorithms. Specifically, we perform the K-means algorithm based on the initial features X, and determine the size of the clusters via finding the elbow point of inertia, a well-adopted clustering metric (Cf. Section 3.3).
- **Multi-Graphs:** If the original graph is composed by multiple sub-graphs, without partition, these sub-graphs can be naturally viewed as different parts of the original graph.

For the corrupt graphs with injected nodes as pseudo anomalies and original nodes as normal ones, we can directly optimize the same context-aware graph contrastive loss function defined in Eq. (1) to train f_{GNN}, which can be further fine-tuned on the target graph if the ground-truth labels are available.

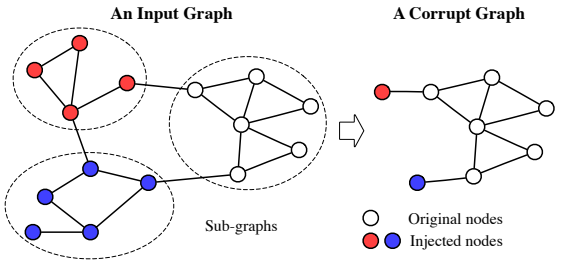


Fig. 3. Illustration of constructing a corrupt graph. A graph is partitioned into multiple sub-graphs. A corrupt graph is constructed from each sub-graph by injecting the nodes outside it.

Connection to Other Graph Pre-Training Frameworks. Existing graph pre-training methods offer limited benefits to anomaly detection. For example, GraphSAGE [13] and GAE [22] reconstruct the adjacency matrix under the local proximity assumption, which is hurt by the suspicious links connecting nodes with opposite labels. GCC [35] maximizes the consistency between the sampled ego-networks of the same node compared with those sampled from different nodes, such feature consistency assumption cannot explicitly help distinguish the abnormal nodes from the normal ones. DGI [44], which contrasts a node with the whole graph, is akin to the contrastive objective function of GCCAD-pre, while the negative instances are addressed differently. In DGI, given a graph as the context, the positive and negative nodes are convolved and represented in independent graphs before contrasting. But in GCCAD-pre, the negative nodes sampled from other graphs are injected and may link to the normal nodes in the corrupted graph, which increases the difficulty to distinguish the normal and abnormal nodes during training, and thus improves the discrimination ability of the model. Essentially, DGI focuses more on representing the normal pattern while GCCAD-pre additionally reinforces the differences of the anomalies from the normal pattern.

2.4 The GNN Encoder of GCCAD

2.4.1 Overview

To optimize the context-aware contrastive objective in Eq. (1), we design the edge update, node update, and graph update modules in the GNN encoder of GCCAD. Edge update, as the first step, is to estimate the likelihood of each link being a suspicious one and then remove the suspicious ones. Then, node update is to update the node embeddings by message passing on the updated adjacency matrix. Finally, the global context is updated via graph update. The three modules are executed at each graph convolution layer successively. After L-layer graph convolutions, the context-aware contrastive loss function is optimized. The framework is illustrated in Figure 4.

2.4.2 Edge Update.

We define suspicious links as the ones that connect nodes with opposite labels. For example, almost every fraudster in business websites can be linked to some benign users by the commonly rated products. Such suspicious links violate the homophily assumption between neighbors—two neighboring nodes tend to have similar features and same labels—and impact the performance of the traditional message-passing along with the links.

Although many attempts have been made to detect such suspicious links [10], [46], [59], they predict the likelihood of a link only based on the linked node features regardless of the node labels. Differently, we additionally model the relative distance between nodes and the global context $\boldsymbol{h}-\boldsymbol{q}$. The promise here is that the distance can serve as implicit labels for guiding the model to remove suspicious links — those connected by the nodes with distinguished distance to the global context.

Context-Aware Link Predictor. The link predictor accepts the representations of two linked nodes $\boldsymbol{h}_i^{(l-1)}$ and $\boldsymbol{h}_j^{(l-1)}$ as the input, and then estimates the linkage likelihood between node v_i and v_j by:

$$p_{ij}^{(l)} = \text{MLP}\left((\boldsymbol{h}_i^{(l-1)} - \boldsymbol{h}_j^{(l-1)}) \oplus (\boldsymbol{h}_i^{(l-1)} - \boldsymbol{q}^{(l-1)}) \oplus (\boldsymbol{h}_j^{(l-1)} - \boldsymbol{q}^{(l-1)})\right), \tag{10}$$

where $\oplus$ is the concatenation operator and MLP is a projection layer with a sigmoid activation function to normalize the likelihood into [0,1]. $(\boldsymbol{h}_i^{(l-1)} - \boldsymbol{h}_j^{(l-1)})$ explicitly denotes the similarity of two nodes, while $(\boldsymbol{h}_i^{(l-1)} - \boldsymbol{q}^{(l-1)})$, the distance from v_i to the global context, implicitly indicates that two nodes—even if their local features are slightly different—can be highly probably linked given their relative positions to the global context are similar. The ablation studies in Section 3.2 empirically proves the effectiveness of this context-aware link predictor.

To accelerate the optimization of the link predictor, we optimize the prediction loss in addition to the contrastive objective in Eq. (1). Specifically, we treat the links between normal nodes as positive ones and the suspicious links as negative ones. The prediction loss is defined as

$$\mathcal{L}_{\text{link}} = \mathbb{E}\left[\sum_{i,j:y_i=y_j=0} -\log p_{ij}^{(l)} - \sum_{i,j:y_i\neq y_j=0} \left(1 - \log p_{ij}^{(l)}\right)\right], \tag{11}$$

which acts as a structural constraint to directly maximize the probabilities of normal links and minimize the probabilities of suspicious ones.

Gumbel-softmax Link Remover. We remove the suspicious links to reduce their negative influence thoroughly by employing the Bernoulli approximation, the binary case of the Gumbel-softmax reparameterization trick [19], [32] to resolve the discrete non-differentiable problem. Specifically, we define the indicator matrix $I \in \mathbb{R}^{N\times N}$, and for the link likelihood $p_{ij}^{(l)}$, we decide whether to keep the link $(I_{ij}^{(l)} = 1)$ or not $(I_{ij}^{(l)} = 0)$ by the following equation:

$$I_{ij}^{(l)} = \text{Bernoulli}\left[\frac{1}{1+\exp\left(-(\log p_{ij}^{(l)} + \varepsilon)/\lambda\right)}\right], \tag{12}$$

where Bernoulli (p) is the Bernoulli approximation with the probability p to keep the link, λ is the temperature

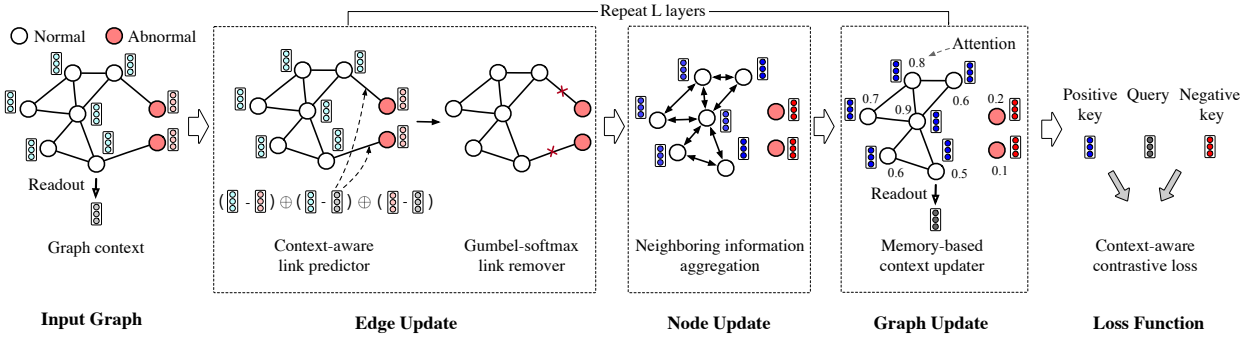


Fig. 4. The framework of GCCAD. At each layer, GCCAD estimates the likelihood for each edge being suspicious and then removes the most suspicious ones with the adjacency matrix updated. Its next step is to update the node embeddings by message passing based on the updated adjacency matrix. Finally it updates the global context. After L-layer graph convolutions, the context-aware contrastive loss function is optimized.

hyper-parameter to control the sharpness of the sampling process (the lower λ is, the more links will be kept), and $\varepsilon \sim$ Gumbel $(0, 1)$ is the Gumbel noise for reparameterization. In the forward process, we sample according to Eq. (12) to obtain the indicator $I_{ij}^{(l)}$. In the backward process, a straight-through gradient estimator [4] is employed to pass the gradients to the relaxed value $p_{ij}^{(l)}$ instead of the discrete value $I_{ij}^{(l)}$.

Edge Residual. Recently, RealFormer [15] verifies the effect of the residual connection of attention scores. Inspired by this, we perform the residual connection of the original edge likelihood and the estimated edge likelihood to smooth the edge evolution as follows:

$$A_{ij}^{(l)} = \left(\alpha A_{ij}^{(l-1)} + (1-\alpha)p_{ij}^{(l)}\right) \odot I_{ij}^{(l)}, \tag{13}$$

where α is a learnable parameter to balance the weight from the last layer and the estimated weight p_{ij}, and $\odot$ denotes the Hadamard product.

2.4.3 Node Update.

Updating node embeddings can be divided into an AGGREGATION and a COMBINE operation:

$$\boldsymbol{a}_i^{(l)} = \text{AGGREGATION}(\{\boldsymbol{h}_j^{(l-1)} : A_{ij}^{(l)} > 0\}), \tag{14}$$
$$\boldsymbol{h}_i^{(l)} = \text{COMBINE}(\boldsymbol{h}_i^{(l-1)}, \boldsymbol{a}_i^{(l)}), \tag{15}$$

where Eq. (14) is to aggregate the neighboring messages of last layer based on the updated adjacency matrix and Eq.(15) is to combine the aggregated information with the concerned node. Generally, the two operations are flexible enough to fit any GNNs. Experiments in Section 3 have investigated several concrete functions.

2.4.4 Graph Update.

Straightforwardly, we can perform sum, max, or average pooling of all the nodes to update the global context. However, they do not distinguish the normal and abnormal nodes, which result in the inaccurate global context. To alleviate it, we introduce a memory buffer $\boldsymbol{m}$ to register the global context $\boldsymbol{q}^{(l-1)}$ of the last layer, based on which we calculate the contribution of each node to the global context.

Algorithm 1: GCCAD or GCCAD-pre

Input: A set of labeled/pseudo-labeled graphs $\{G_i\}_{i=1}^{M}$. Each graph G_i is consist of the adjacency matrix $A \in \mathbb{R}^{N\times N}$, and feature matrix $X \in \mathbb{R}^{N\times d}$. Also the learning rate η, an L-layers GNNs encoder: $f_{\text{GNN}}(X, A, W)$.
Output: Learned parameter W of f_{GNN}, where $W = \{W_{\text{edge}}, W_{\text{node}}, W_{\text{global}}\}$

Initialize the global context $\boldsymbol{q}^{(0)} = \frac{1}{N}(\sum_{j=1}^{N} \boldsymbol{x}_j)$;
Initialize the trainable parameter W of f_{GNN};
repeat
  foreach G_i **do**
    for l *from 1 to* L **do**
      Edge update to get the adjacency matrix: $A^{(l)} = f_{\text{edge}}\left(H^{(l-1)}, A^{(l-1)}, \boldsymbol{q}^{(l-1)}, W_{\text{edge}}\right)$.
      Node update to get the node embeddings: $H^{(l)} = f_{\text{node}}\left(H^{(l-1)}, A^{(l)}, W_{\text{node}}\right)$.
      Graph update to get the global context: $\boldsymbol{q}^{(l)} = f_{\text{graph}}(H^{(l)}, \boldsymbol{q}^{(l-1)}, W_{\text{global}})$
    Compute the loss function in Eq. (17).
    Update W by $W = W - \eta\nabla_W \mathcal{L}$;
until *Converges*;

The assumption is a node that is deviated from the current global context should be weakened in the next update step. Such the memory-based mechanism has been successfully applied to general computation machines [3], [51]. Formally,

$$s_i^{(l)} = \text{cosine}\left(\boldsymbol{h}_i^{(l)}, \boldsymbol{m}\right), \quad \alpha_i^{(l)} = \frac{\exp(s_i^{(l)})}{\sum_{j=1}^{N}\exp(s_j^{(l)})}, \tag{16}$$
$$\boldsymbol{q}^{(l)} = \sum_{i=1}^{N} \alpha_i^{(l)} \cdot \boldsymbol{h}_i^{(l)}, \quad \boldsymbol{m} = \boldsymbol{q}^{(l)}.$$

First, we compute the attention $\alpha_i^{(l)}$ for each node v_i based on its cosine similarity with the memory vector. Then we update the global context $\boldsymbol{q}^{(l)}$ by weightedly aggregating different nodes and update the memory vector $\boldsymbol{m}$.

2.5 Training and Inference

In each epoch, we optimize both the context-aware contrastive loss in Eq. (1) and the link predictor constraint loss in Eq. (11) as follows:

TABLE 1
Data statistics. We present the total and the average number of the nodes, edges in all the graphs. The number of relation contained in the links. Concentration of a graph is calculated as the average cosine similarity of all pairs of nodes, wherein a node is instantiated by its eigenvector. Concentration of a dataset is the average concentration of all the graphs.

Datasets	Multi-graph		Single-graph	
	AMiner	MAS	Alpha	Yelp
#Nodes	192,352	117,974	3,783	45954
(Normal, Abnormal%)	(90.5, 9.5)	(84.4, 15.6)	(3.6, 2.7)	(85.4, 14.5)
#Total Links	12,669,793	2,543,833	24,186	3,846,979
#Graphs	1,104	2,098	1	1
Average #nodes	174	56	-	-
Average #links	11,476	1,212	-	-
#Relation	3	2	2	3
Concentration	0.0063	0.0075	0.0088	0.0003
Normal links %	97.23	97.00	-	77.30

TABLE 2
Performance of GCCAD compared with the baseline models (%) .

Model	Multi-graph				Single-graph			
	AMiner		MAS		Alpha		Yelp	
	AUC	MAP	AUC	MAP	AUC	MAP	AUC	MAP
Graph Neural Networks								
GCN	75.68	59.28	78.44	63.89	82.12	78.09	70.85	29.18
GAT	74.97	59.87	78.25	62.75	78.45	76.32	75.28	32.11
GSAGE	73.01	60.51	76.55	62.12	79.23	75.14	76.34	34.12
GIN	74.28	57.75	75.62	59.96	87.39	86.79	76.67	34.60
Graph-based Anomaly Detection Models								
LogisticReg	62.49	33.68	73.67	56.64	71.88	72.43	67.74	26.27
GeniePath	76.18	50.67	81.93	65.66	75.02	68.89	76.78	33.67
GraphConsis	71.37	52.21	76.67	60.25	86.99	86.63	70.27	26.64
CARE-GNN	77.74	55.38	80.01	66.98	84.26	85.20	77.14	36.84
GCCAD	**89.84**	**78.73**	**87.62**	**75.18**	**90.53**	**91.20**	**79.64**	**37.15**

$$\mathcal{L} = \mathcal{L}_{\text{con}} + \lambda \mathcal{L}_{\text{link}}, \tag{17}$$

where λ is a hyper-parameter that balances the two losses. We empirically set λ as 0.2 in our model. Algorithm 1 outlines the training process. During inference, for node $v_i \in G$ to be predicted, instead of directly predicting a binary label, we estimate its abnormality score by computing the cosine similarity between its embedding $\boldsymbol{h}_i^{(L)}$ and the graph embedding $\boldsymbol{q}^{(L)}$ of the final layer (the lower, the more anomalous).

2.6 Complexity

The time complexity of GCCAD is the same order of magnitude with the vanilla GNN. Because on top of the node update step of GNN, GCCAD adds two additional efficient steps, edge update and graph update. The time complexity of edge update in layer k is $\mathcal{O}(D_k E)$, where E is the number of edges and D_k is the embedding size in layer k. The time complexity of graph update in layer k is $\mathcal{O}(D_k N)$, where N is the number of nodes. Since the embedding size is far smaller than the number of nodes or edges, the time complexity of GCCAD grows linearly with the graph scale, which is the same as the vanilla GNN.

3 Experiment

In this section, we perform two major experiments to verify the performance of the supervised GCCAD and the unsupervised GCCAD-pre respectively on two genres of anomaly detection applications. All the codes and the datasets are online now[7].

3.1 Experimental Setup

Datasets. We evaluate the proposed models for detecting the wrongly assigned papers in researchers' profiles on two academic datasets and detecting users who give fraudulent ratings on two business websites. The academic datasets are both multi-graph datasets and the business datasets are both single-graph datasets. Table 1 shows the statistics.

- **AMiner:** is a free online academic system[8] collecting over 100 million researchers and 260 million publications [42]. We extract 1,104 online researchers' profiles, then for each profile, we build a graph by adding papers as nodes and creating a link between two papers if they share the same coauthors, venues or organizations[9]. The wrongly assigned papers are provided by human annotators.
- **MAS:** is the Microsoft academic search system[10] containing over 19 million researchers and 50 million publications [40]. 2,098 graphs are extracted and constructed similarly as AMiner, and the ground-truth labels are provided by KDD cup 2013 [38].
- **Alpha [25]:** is a Bitcoin trading platform. A single graph is created by adding users as nodes and the rating relationships between users as links. Benign users are the platform's founders or users who are rated positively. Fraudsters are the users who are rated negatively by the benign users.
- **Yelp [36]:** is a platform for users to rate hotels and restaurants. A single graph is created by adding spam (abnormal) and legitimate (normal) reviews filtered by Yelp as nodes. The links are created between two reviews if they are posted by the same user, on the same product with the same rating, or in the same month.

For the two multi-graph academic datasets, we use the paper title embedded by BERT as the initial feature for each node. For Yelp, we leverage a sparse matrix of 100-dimension Bag-of-words initial features for each node [10]. For Alpha, since we do not have the side information of nodes, we conduct eigen-decomposition on the graph Laplacian s.t. $I - D^{-1/2}AD^{-1/2} = U\Lambda U^{\top}$ with I as the identity matrix, D as the degree matrix, and use the top eigenvectors in U [45] as the initial 256-dimensional features for each node.

Evaluation Metrics. We adopt Area Under ROC Curve (AUC), broadly adopted in anomaly detection [10], [31], and Mean Average Precision (MAP), which pays more attention to the rankings of the anomalies, as the evaluation metrics.

7. https://github.com/allanchen95/GCCAD

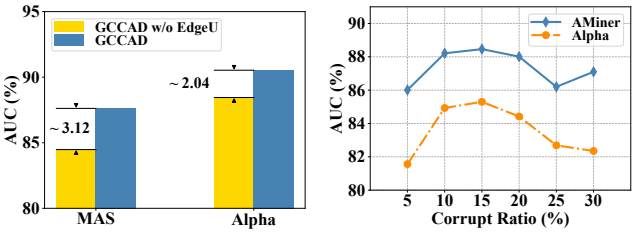
8. www.aminer.org

9. The profiles of the system are created automatically and cannot avoid the mistakes.

10. cn.bing.com/academic

TABLE 3
Ablation Studies of GCCAD(%). We adopt the best node update strategy, i.e., GCN for Multi-graph datasets and GIN for Single-graph datasets.

Model	Multi-graph (GCN)				Single-graph (GIN)			
	AMiner		MAS		Alpha		Yelp	
	AUC	MAP	AUC	MAP	AUC	MAP	AUC	MAP
GCCAD	**89.84**	**78.73**	**87.62**	**75.18**	**90.53**	**91.20**	**79.64**	**37.15**
Variant Loss Functions								
+ CE_loss	80.30	61.02	81.53	67.42	88.64	89.10	79.18	37.05
Variant Edge Update Strategies								
- LP_constraint	89.36	78.57	87.25	74.56	89.70	90.55	78.63	35.62
- Global Info.	88.67	77.13	86.78	73.48	89.93	90.63	78.61	35.20
- Edge Update	88.21	76.97	86.34	71.54	89.69	89.86	78.51	35.19
Variant Node Update Strategies								
$\text{GCCAD}_{\text{GCN}}$	**89.84**	**78.73**	**87.62**	**75.18**	85.07	87.09	73.33	30.64
$\text{GCCAD}_{\text{GAT}}$	89.00	77.98	86.30	72.75	83.26	86.37	72.87	29.91
$\text{GCCAD}_{\text{GSAGE}}$	87.84	75.26	82.86	68.52	86.54	87.07	76.52	33.28
$\text{GCCAD}_{\text{GIN}}$	89.04	77.02	83.39	67.46	**90.53**	**91.20**	**79.64**	**37.15**
Variant Graph Update Strategies								
+ Avg Pooling	89.01	77.66	86.39	72.73	89.02	89.57	78.70	34.55
+ Sum Pooling	86.95	72.09	86.45	73.98	88.01	86.69	76.79	34.11
+ Max Pooling	80.91	61.97	84.06	70.52	89.25	88.56	78.54	33.96



(a) GCCAD vs. w/o EdgeU (b) MAP vs. Corrupt Ratio

Fig. 5. We study (a) the performance gain of the supervised GCCAD obtained by edge update step on the filtered Alpha and MAS datasets; (b) the best AUC of GCCAD-pre without fine-tuning when varying the corrupt ratio, i.e., the ratio of the injected abnormal nodes, on AMiner and Alpha.

3.2 Evaluation of GCCAD

Baselines. We compare GCCAD with four general GNNs models, including GCN [21], GAT [43], GraphSAGE [13], and GIN [54], and three specific GNN models for anomaly detection, including GeniePath [29], GraphConsis [31], and CARE-GNN [10], which are further introduced in Section 4. Other GNN-based anomaly detection models such as GAS [26], FdGars [47], Semi-GNN [46], and Player2Vec [59], that are empirically proven to be less useful than the adopted baselines, are ignored in the experiments. We additionally compare with logistic regression by injecting top eigenvectors as features to perform node classification.

Setup. For the multi-graph datasets, i.e., AMiner and MAS, we extract 70% graphs as the training set, 10% as the validation set, and 20% as the test set. During training, we perform GCCAD on each graph following Algorithm 1. For testing, we rank nodes based on the cosine similarities between their node embeddings and the corresponding graph context in ascending order, then we evaluate AUC and MAP in each test graph and average the metrics of all the test graphs as the final metrics. For the single-graph datasets, i.e. Alpha and Yelp, likewise, we extract 70% of the labeled nodes from the single graph as the training data, 10% as the validation set, and 20% as the test set. We perform GCCAD on the single graph, and evaluate AUC and MAP for the ranked nodes in the test data. Note that, for GraphConsis [31] and CARE-GNN [10] whose inputs are the graph with multi-relation links, we use its multi-relation information, while GCCAD merges the multi-relation links between two nodes into the single-relation links. For all experiments, we run 5 trials and report the mean results.

Implementation. For GCCAD , the number of layers L is empirically set as 2, the temperature hyperparameter τ is set as 0.1 in Eq. (1), and λ is set as 0.6 in Eq. (12). We use Adam with learning rate 1×10^{-3} for training. The learning rate is set as the initial value over the first 10% steps, and then linearly decay to 1/10 of the initial value.

For the GCN, GAT, GraphSAGE, and GIN, We leverage the implementations provided by Pytorch Geometric[11], and set the number of convolutional layers as 2 for all general GNNs. For GeniePath, GraphConsis, and CARE-GNN, we use the authors' official codes with the same training settings. The data format is transformed appropriately to fit their settings. All models are running on Python 3.7.3, 1 NVIDIA Tesla V100 GPU, 512GB RAM, Intel(R) Xeon(R) Gold 5120 CPU @ 2.20GHz.

Overall Results. Table 2 presents the performance of GCCAD and all the comparison methods on four datasets. We can see that, GCCAD substantially improves over all other baselines, +2.5-27.35% in AUC and +0.31-28.06% in MAP, on all the datasets. Among all the baselines, logistic regression performs the worst as it only leverages the structural information and does not apply graph convolution to integrate neighbor messages. None of the specific graph-based anomaly detection methods can keep the advantage over the general GNNs on all the datasets. For example, GraphConsis [31] performs worse than GIN [54] on AMiner. The highlighted results in the table are from GCCAD, which adopts the context-aware contrastive objective and the three-stage GNN encoder, performing stably the best over all the datasets.

Ablation Studies. To this end, there have two major differences: the context-aware contrastive objective and the three-stage GNN encoder between GCCAD and other baselines. Thus we conduct the ablation studies to verify the efficacy of different components in GCCAD. The four main model variants are presented as follows, and the results are illustrated in Table 3.

- **w/ CE_loss:** Change the contrastive loss in Eq. (1) with cross-entropy objective.
- **Edge Update Strategies:** Change the edge update component in Section 2.4.2 to *w/o LP_constraint* that removes the constraint loss $\mathcal{L}_{\text{link}}$ in Eq (11), *w/o Global Info.* that removes $(\boldsymbol{h}_{(i/j)}^{(l-1)} - \boldsymbol{q}^{(l-1)})$ in Eq.(10), or *w/o Edge Update* that removes the entire edge update step.
- **Node Update Strategies:** Change the node update strategies in Eq. (14) and Eq. (15) to the node update strategies

11. https://github.com/rusty1s/pytorch_geometric

in GCN, GAT, GraphSAGE, or GIN.
- **Graph Update Strategies:** Change the memory-based readout function in Eq.(16) to average pooling (*Ave Pooling*), sum pooling (*Sum Pooling*), and maximal pooling (*Max Pooling*) of all the nodes.

Node Update Strategies. From Table 3, we can see that GCCAD with various node update strategies outperforms other baselines in most cases, which suggests the superiority of context-aware contrastive objective and the three-stage GNN encoder.

We also observe that different strategies have the performance gap, with less than 6% in AUC and 7% in MAP. On the multi-graph datasets, GCCAD performs the best with the mean AGGREGATION and the concatenation COMBINE strategies in GCN [21], while on the single-graph datasets, it performs the best with the sum AGGREGATION and the concatenation COMBINE strategies in GIN [54]. Obviously, the trend of performance changes over different node update strategies is akin to that of general GNNs, as shown at the top of Table 2. Thus we conjecture the instability is mainly due to the intrinsic traits of the node update strategy in GNNs.

Thus, unless stated otherwise, we only conduct experiments of GCCAD with the best node update strategy, that is, GCN on the multi-graph datasets and GIN on the single-graph datasets, in the remaining parts.

The Contrastive Loss Function. From Table 3, we observe that the performance gain of GCCAD over +CE_loss on the multi-graph datasets (+6.09-9.54% in AUC and +7.76-17.71% in MAP) is much higher than that on the single-graph datasets (+0.46-1.89% in AUC and +0.60-2.10% in MAP). Since the multi-graph datasets correspond to the scenario that the data distributions is dynamically changed in different graphs, while the single-graph datasets correspond to the scenario that the data distributions are relatively static, the results empirically validates Theorem 1 that the context-aware contrastive objective is more resilient than cross-entropy loss when the abnormal or normal node distributions are dynamically changed.

Context-aware GNN Encoder. GCCAD w/ CE_loss outperforms other the state-of-the-art baselines by up to 17.81% AUC, which elucidates the powerful representation capability of the proposed context-aware GNN encoder. The improvements can give credit to two perspectives: the context-aware edge update and the memory-based global update. Thus, We further verify the effectiveness of each component.

Edge Update Strategies. GCCAD w/o LP_constraint performs slightly worse than GCCAD, which denotes link prediction constraint can not only accelerate the optimization of link predictor but also contribute to the overall improvement.

GCCAD w/o Global Info. also underperforms GCCAD, which indicates the context-aware link predictor, that incorporates the relative distance to the global context as the implicit supervision, can benefit the suspicious link prediction.

Compared with the above two variants, GCCAD w/o Edge Update performs the worst (-0.84-1.63% in AUC), which verifies the efficacy of the proposed edge update mechanism. However, we observe the performance enhancement brought by the edge update is not significant. We speculate the norm links which connect the nodes of the same labels occupy the majority of all the links (e.g., 90.5% on AMiner and 84.4% on MAS in Table. 1). Thus, removing the small amount of the suspicious links by the link predictor can only exert limited effect. To verify this, we select the hard instances, i.e., the abnormal nodes that connect at least k[12] normal nodes, from both MAS and Alpha, and re-evaluate the variants on filtered datasets. From Figure 5(a), we can see that GCCAD significantly outperforms w/o Edge Update by 3.12% and 2.04% AUC on the two datasets, respectively. The results indicate that the negative influence of the suspicious links can be effectively reduced by the edge update step.

TABLE 4
Transfer Learning about Cross-Entropy (CE) and Context Contrastive Loss (CL) on AMiner and MAS (AUC %). The model trained on one dataset (top) and evaluate on another one (left).

Model		AMiner	MAS	Performance Drop
AMiner	w/ CL	89.84	87.70	2.14
	w/ CE	80.30	77.05	**3.25**
MAS	w/ CL	84.41	87.62	3.21
	w/ CE	74.15	81.53	**7.38**

Global Update Strategies. The proposed memory-based global context update mechanism performs the best, which improves 0.83-8.93% AUC and 1.07-16.76% MAP over other variant mechanisms. Because the memory that records the global context of the last layer can help estimate the contribution of each node in the current layer, thus it will benefit the generation of precise global context in the current layer.

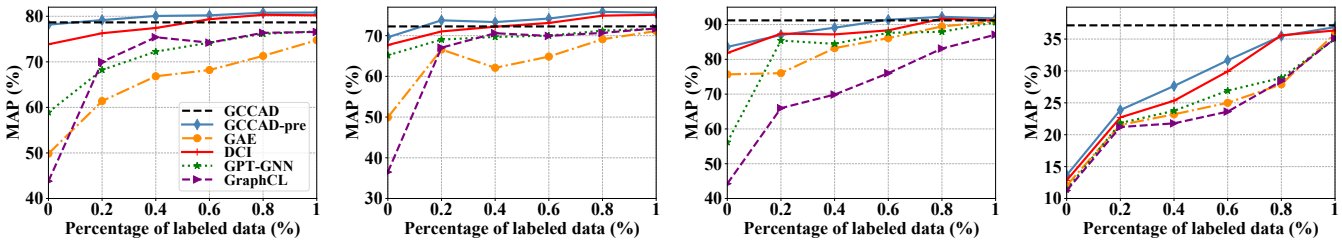
Transfer Learning. To further investigate the robustness of context-aware contrastive loss compared with cross-entropy objective, we train GCCAD and its variant w/ CE_loss on AMiner/MAS and evaluate them on MAS/AMiner[13]. As shown in Table 4, the performance drop of GCCAD (-2.14-3.21% in AUC) is substantially less than that of w/ CE_loss (3.25-7.38% in AUC) under the setting of transfer learning, which shows the superior robustness ability of the context-aware contrastive objective compared with the cross-entropy objective.

3.3 Evaluation of GCCAD-pre

Baselines. We compare GCCAD-pre with the state-of-the-art graph pre-training models, including GAE [22], GPT-GNN [18], GraphCL [58], and a graph pre-training model tailored for anomaly detection, DCI [49], in the unsupervised setting. GAE is to preserve the structural correlations via reconstructing the vertex adjacency matrix. In addition to the structural correlations, GPT-GNN preserves the attribute interplays via predicting the masked attributes. GraphCL maximizes the mutual information of two instances of the same graph obtained by graph data augmentation. DCI [49] is a cluster-based version of DGI, which

12. Empirically, k is set as 1 in MAS and 3 in Alpha respectively.

13. Since the feature initialization process and the embedding dimension of Alpha and Yelp are obviously different from each other, we only conduct the transfer learning experiments between AMiner and MAS.



(a) AMiner (b) MAS (c) Alpha (d) Yelp

Fig. 6. The pre-training and fine-tuning performance given different percentage of the labeled data on four datasets. The horizontal black dashed line represents the performance of the supervised GCCAD.

maximizes the fine-grained mutual information between the embeddings of a cluster and the nodes within it and is proved to be effective for graph anomaly detection. For a fair comparison, all the pre-training models adopt the same GNN encoder as GCCAD.

Datasets. In the online AMiner system, we hide the labeled 1,104 graphs in Table 1 and extract additional 4,800 corrupt graphs from it. According to the multi-graph corrupting strategy in Section 2.3, we first build a single large paper graph for the whole extracted AMiner dataset and then divide it into multiple graphs according to the ownership of papers to different researchers. For each graph, we corrupt it by injecting the papers from other researchers' graphs as the anomalies with a certain corrupt ratio of 15%, i.e., the ratio between the number of injected nodes with the number of existing nodes in the graph. The links in the original graph are kept between the nodes in corrupt graphs. Then we build the MAS corrupting graph dataset in the same way.

For the single-graph datasets, Alpha and Yelp, we first cluster it into K sub-graphs by classic K-means algorithm based on their input features. Then the within-cluster correlation is measured by sum-of-squares criterion, namely inertia[14], estimated by $\sum_{i=0}^{|V|} \min_{\boldsymbol{u}_k \in C} (||\boldsymbol{x}_i - \boldsymbol{u}_k||^2)$, where $\boldsymbol{u}_k$ is the embeddings of k-th cluster center and C is K disjoint clusters. To automatically set a proper K, we leverage the Elbow Methods to choose the elbow point of inertia, i.e, to locate the optimal K^* at which the trend of inertia changes from steep to stable (Cf. Figure 7). Finally, the graph size K is set as 20 on Alpha and 30 on Yelp.

Setup. Without any labeled data (0% labeled data), we train GCCAD-pre and the baselines on the corrupt graphs and evaluate them on the same test set as the supervised GCCAD. Then we also explore the few-shot learning settings, i.e., we further fine-tune the pre-trained models when given a proportion of labeled data.

Implementation. For GCCAD-pre, we follow the same setting as GCCAD. For GAE, DCI, GPT-GNN, and GraphCL, we use the authors' official codes with the same training settings. Note that, for GraphCL, we try all the graph augmentation methods defined in the paper and and select the sub-graph augmentation that achieves the best performance in the test set without fine-tuning.

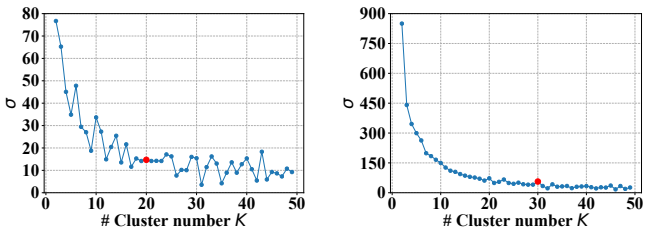
14. https://scikit-learn.org/stable/modules/clustering.html#k-means

Overall Results. Figure 6 shows the pre-training and fine-tuning performance of all the comparison methods given different percentage of labeled data on four datasets. From the results, we can see that GCCAD-pre (blue line) fine-tuned on 0%, 10%, and 60% labeled data is comparable with the fully-supervised GCCAD (black dashed line) on AMiner, MAS, and Alpha respectively. What's more, when GCCAD-pre is fined-tuned on all the labeled data, it even outperforms GCCAD by 1.72%, 2.19%, and 1.04% in AUC on AMiner, MAS, and Alpha respectively, which demonstrates the effectiveness of the proposed pre-training framework.

Our model also outperforms all the baselines when whatever percentage of labeled data is provided. GAE and GPT-GNN focus on reconstructing the links in the graphs. GraphCL aims to capture the normal distribution of the whole graphs. The objective of them is a little far away from anomaly detection. DCI performs the best among all the baselines. Given the global context as the query, DCI aims to contrast between the nodes within the sub-graph (normal nodes) and those from other sub-graphs (abnormal nodes). The objective is similar to GCCAD-pre, but different from the corrupted graphs by GCCAD-pre, the anomalies in DCI are thoroughly disconnected from the normal nodes and are independently embedded in other graphs, which may reduce the learning difficulty of the pseudo labels compared with the ground-truth labels.

We also observe that on Yelp, none of the pre-training models achieve prominent performance without any labeled data. On one hand, as shown in Table 1, because of the smallest concentration, Yelp is the most diverse dataset, which prevents GCCAD-pre, DCI, and GraphCL from discovering the proper normal pattern from the whole graph. On the other hand, since the ratio of the suspicious links on Yelp, 22.70%, is the largest among the datasets, GAE and GPT-GNN that target at preserving the link homophily will wrongly reconstruct those noisy links. However, GCCAD-pre still performs best compared with other pre-training models whatever percentage of labeled data is provided, which implies the ability of GCCAD-pre to distinguish the normal nodes from the abnormal ones.

Corrupt Ratio. Figure 5(b) presents the correlation between the ratio of the injected nodes (corrupt ratio) and the performance of GCCAD-pre on AMiner and Alpha without fine-tuning. The results show that the best performance is achieved when about 15% of abnormal nodes are injected. The corrupt ratio is set in the same way as other datasets.



(a) Alpha (b) Yelp

Fig. 7. The correlation between the graph (cluster) number K and the inertia gap $\sigma(K)$ between K and $K+1$ on Alpha (a) and Yelp (b). Red point denotes the K^* that achieves the best performance of GCCAD-pre without fine-tuning.

Clustering Number. Figure 7 presents the correlation between the graph (cluster) number K and the inertia gap $\sigma(K)$ between K and $K+1$ on Alpha and Yelp. The optimal K^* that achieves the best performance of GCCAD-pre without fine-tuning are marked as red in the Figure. We observe that GCCAD-pre performs the best when K^* (red) is approximately around the elbow point where the trend of inertia changes from steep to stable. Although we choose the elbow points on Yelp, Yelp is so diverse that none of the graph pre-training methods can achieve desired performance.

4 Related Work

4.1 Graph-based Anomaly Detection

Graph neural networks have been studied to detect anomalies in various domains, such as detecting review spams in business websites [10], [26], [28], [31], rumors in social media [5], [52], [55], fake news [33], [37], financial fraud [29], [30], [46], insurance fraud [27], and bot fraud [57]. Most of them target how to design a proper aggregator that can distinguish the effects of different neighbors and reduce the inconsistency issue [31] during message passing. For example, GAS [26] adopts the vallina GCN [21]. SemiGNN [46] and Player2Vec [59] apply attention mechanisms to assign low weights to suspicious links. To thoroughly reduce the negative influence of the suspicious links, several attempts [12], [31] have been made to remove the suspicious links before graph convolution. CARE-GNN [10] further adopts reinforcement learning to sample links according to the suspicious likelihood. None of them are aware of the limitations caused by the objective function. To our knowledge, we are the first to change the commonly-used binary classification into the graph contrastive coding paradigm.

4.2 Graph Pre-training Schemes

With the advances of self-supervised (pre-training) techniques in visual representation learning [7], [14], graph pre-training schemes have also attracted increasing attention. A naive GNN pre-training scheme is to reconstruct the vertex adjacency matrix, and GAE [22] and GraphSAGE [13] are two representative models. In addition to preserve the structure homophily, GPT-GNN [18] preserves the attribute homophily by predicting the masked node attributes. Motivated by [16], DGI [44] and Infograph [41] have been proposed to maximize the mutual information between the embeddings of the entire graph and the node within it. GraphCL [58] maximizes the mutual information between the embeddings of two graph instances, which are obtained from the same graph via graph data augmentation. Later, GCC [35] shrinks the contrast between graphs into that between ego-networks, where the ego-network instances are obtained via random walk with start from the concerned node. All of them are proven to be useful for the downstream node classification task, but are not specifically proposed to solve the anomaly detection problem. DCI [49] is a cluster-based version of DGI, which maximizes the fine-grained mutual information between the embeddings of a cluster and the nodes within it. DCI is proposed for graph anomaly detection. However, it is thoroughly unsupervised model and different from the corrupted graphs by GCCAD-pre, the anomalies in DCI are thoroughly disconnected from the normal nodes, which may reduce the learning difficulty of the pseudo labels.

5 Conclusions

This paper proposes GCCAD, a graph contrastive coding model for anomaly detection on graphs. Instead of directly classifying the node labels, GCCAD contrasts the abnormal nodes with normal ones in terms of their distance to the global context of the graph. We further extend GCCAD to an unsupervised version by designing a graph corrupting strategy to generate the synthetic node labels. To achieve the contrastive objective, we design a three-stage GNN encoder to infer and remove the suspicious links during message passing, as well as to learn the global context. The experimental results on four real-world datasets for anomaly detection demonstrate that GCCAD significantly outperforms the baselines and GCCAD-pre without any fine-tuning can achieve comparable performance with the fully-supervised version on the two academic datasets. We will continue to study more promising pre-training strategies for detecting anomalies on Yelp, the most diverse dataset among all the evaluated ones.

References

[1] I. Ahmad and P.-E. Lin. A nonparametric estimation of the entropy for absolutely continuous distributions (corresp.). *IEEE Trans. Inf. Theory*, 22(3):372–375, 1976.
[2] L. Akoglu, H. Tong, and D. Koutra. Graph based anomaly detection and description: a survey. *DMKD'15*, 29(3):626–688, 2015.
[3] D. Bahdanau, K. H. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR'15*, 2015.
[4] Y. Bengio, N. Léonard, and A. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.
[5] T. Bian, X. Xiao, T. Xu, P. Zhao, W. Huang, Y. Rong, and J. Huang. Rumor detection on social media with bi-directional graph convolutional networks. In *AAAI'20*, volume 34, pages 549–556, 2020.
[6] M. Boudiaf, J. Rony, I. M. Ziko, E. Granger, M. Pedersoli, P. Piantanida, and I. B. Ayed. A unifying mutual information view of metric learning: cross-entropy vs. pairwise losses. In *ECCV'20*, pages 548–564. Springer, 2020.
[7] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *ICML'20*, pages 1597–1607. PMLR, 2020.
[8] L. Cui, H. Seo, M. Tabar, F. Ma, S. Wang, and D. Lee. Deterrent: Knowledge guided graph attention network for detecting healthcare misinformation. In *KDD'20*, pages 492–502, 2020.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT'19*, pages 4171–4186, 2019.

[10] Y. Dou, Z. Liu, L. Sun, Y. Deng, H. Peng, and P. S. Yu. Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. In *CIKM'20*, pages 315–324, 2020.

[11] T. Fawcett and F. J. Provost. Combining data mining and machine learning for effective user profiling. In *KDD'96*, pages 8–13, 1996.

[12] L. Franceschi, M. Niepert, M. Pontil, and X. He. Learning discrete structures for graph neural networks. In *ICML'19*. PMLR, 2019.

[13] W. L. Hamilton, Z. Ying, and J. Leskovec. Inductive representation learning on large graphs. In *NIPS'17*, pages 1024–1034, 2017.

[14] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR'20*, pages 9729–9738, 2020.

[15] R. He, A. Ravula, B. Kanagal, and J. Ainslie. Realformer: Transformer likes residual attention. *arXiv preprint arXiv:2012.11747*, 2020.

[16] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio. Learning deep representations by mutual information estimation and maximization. In *ICLR'18*, 2018.

[17] B. Hooi, H. A. Song, A. Beutel, N. Shah, K. Shin, and C. Faloutsos. Fraudar: Bounding graph fraud in the face of camouflage. In *KDD'16*, pages 895–904, 2016.

[18] Z. Hu, Y. Dong, K. Wang, K.-W. Chang, and Y. Sun. Gpt-gnn: Generative pre-training of graph neural networks. In *KDD'20*, pages 1857–1867, 2020.

[19] E. Jang, S. Gu, and B. Poole. Categorical reparameterization with gumbel-softmax. *ICLR'17*, 2016.

[20] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan. Supervised contrastive learning. *NIPS'20*, 33, 2020.

[21] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. 2016.

[22] T. N. Kipf and M. Welling. Variational graph auto-encoders. *NIPS'16*, 2016.

[23] A. Kraskov, H. Stögbauer, and P. Grassberger. Estimating mutual information. *Physical review E*, 69(6):066138, 2004.

[24] S. Kumar, B. Hooi, D. Makhija, M. Kumar, C. Faloutsos, and V. Subrahmanian. Rev2: Fraudulent user prediction in rating platforms. In *WSDM'18*, pages 333–341, 2018.

[25] S. Kumar, F. Spezzano, V. Subrahmanian, and C. Faloutsos. Edge weight prediction in weighted signed networks. In *ICDM'16*, pages 221–230. IEEE, 2016.

[26] A. Li, Z. Qin, R. Liu, Y. Yang, and D. Li. Spam review detection with graph convolutional networks. In *CIKM'19*, 2019.

[27] C. Liang, Z. Liu, B. Liu, J. Zhou, X. Li, S. Yang, and Y. Qi. Uncovering insurance fraud conspiracy with network learning. In *SIGIR'19*, pages 1181–1184, 2019.

[28] Y. Liu, X. Ao, Z. Qin, J. Chi, J. Feng, H. Yang, and Q. He. Pick and choose: A gnn-based imbalanced learning approach for fraud detection. In *WWW'21*, pages 3168–3177, 2021.

[29] Z. Liu, C. Chen, L. Li, J. Zhou, X. Li, L. Song, and Y. Qi. Geniepath: Graph neural networks with adaptive receptive paths. In *AAAI'19*, volume 33, pages 4424–4431, 2019.

[30] Z. Liu, C. Chen, X. Yang, J. Zhou, X. Li, and L. Song. Heterogeneous graph neural networks for malicious account detection. In *CIKM'18*, pages 2077–2085, 2018.

[31] Z. Liu, Y. Dou, P. S. Yu, Y. Deng, and H. Peng. Alleviating the inconsistency problem of applying graph neural network to fraud detection. In *SIGIR'20*, pages 1569–1572, 2020.

[32] C. J. Maddison, A. Mnih, and Y. W. Teh. The concrete distribution: A continuous relaxation of discrete random variables. *ICLR'17*, 2016.

[33] V. Nguyen, K. Sugiyama, P. Nakov, and M. Kan. FANG: leveraging social context for fake news detection using graph representation. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*, pages 1165–1174, 2020.

[34] A. v. d. Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[35] J. Qiu, Q. Chen, Y. Dong, J. Zhang, H. Yang, M. Ding, K. Wang, and J. Tang. Gcc: Graph contrastive coding for graph neural network pre-training. In *KDD'20*, pages 1150–1160, 2020.

[36] S. Rayana and L. Akoglu. Collective opinion spam detection: Bridging review networks and metadata. In *KDD'15*, 2015.

[37] Y. Ren, B. Wang, J. Zhang, and Y. Chang. Adversarial active learning based heterogeneous graph neural network for fake news detection. In *ICDE'20*, 2020.

[38] S. B. Roy, M. De Cock, V. Mandava, S. Savanna, B. Dalessandro, C. Perlich, W. Cukierski, and B. Hamner. The microsoft academic search dataset and kdd cup 2013. In *Proceedings of the 2013 KDD cup 2013 workshop*, pages 1–6, 2013.

[39] N. Ruchansky, S. Seo, and Y. Liu. Csi: A hybrid deep model for fake news detection. In *CIKM'17*, pages 797–806, 2017.

[40] A. Sinha, Z. Shen, Y. Song, H. Ma, D. Eide, B.-J. Hsu, and K. Wang. An overview of microsoft academic service (mas) and applications. In *WWW'15*, pages 243–246, 2015.

[41] F.-Y. Sun, J. Hoffmann, V. Verma, and J. Tang. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. In *ICLR'20*, 2020.

[42] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su. Arnetminer: extraction and mining of academic social networks. In *KDD'08*, pages 990–998, 2008.

[43] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio. Graph attention networks. 2017.

[44] P. Velickovic, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm. Deep graph infomax. In *ICLR'19*, 2019.

[45] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[46] D. Wang, J. Lin, P. Cui, Q. Jia, Z. Wang, Y. Fang, Q. Yu, J. Zhou, S. Yang, and Y. Qi. A semi-supervised graph attentive network for financial fraud detection. pages 598–607, 2019.

[47] J. Wang, R. Wen, C. Wu, Y. Huang, and J. Xion. Fdgars: Fraudster detection via graph convolutional networks in online app review system. In *WWW'19*, pages 310–316, 2019.

[48] T. Wang and P. Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *ICML'20*, pages 9929–9939. PMLR, 2020.

[49] Y. Wang, J. Zhang, S. Guo, H. Yin, C. Li, and H. Chen. Decoupling representation learning and classification for gnn-based anomaly detection. In *SIGIR'21*, pages 1239–1248, 2021.

[50] C. Wei, K. Sohn, C. Mellina, A. Yuille, and F. Yang. Crest: A class-rebalancing self-training framework for imbalanced semi-supervised learning. In *ICCV'21*, pages 10857–10866, 2021.

[51] J. Weston, S. Chopra, and A. Bordes. Memory networks. *arXiv preprint arXiv:1410.3916*, 2014.

[52] Y. Wu, D. Lian, Y. Xu, L. Wu, and E. Chen. Graph convolutional networks with markov random field reasoning for social spammer detection. In *AAAI'20*, volume 34, pages 1054–1061, 2020.

[53] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR'18*, pages 3733–3742, 2018.

[54] K. Xu, W. Hu, J. Leskovec, and S. Jegelka. How powerful are graph neural networks? In *ICLR'18*, 2018.

[55] X. Yang, Y. Lyu, T. Tian, Y. Liu, Y. Liu, and X. Zhang. Rumor detection on social media with graph structured adversarial learning. In *IJCAI'20*, pages 1417–1423, 2020.

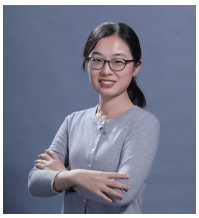
[56] Y. Yang and Z. Xu. Rethinking the value of labels for improving class-imbalanced learning. *arXiv preprint arXiv:2006.07529*, 2020.

[57] T. Yao, Q. Li, S. Liang, and Y. Zhu. Botspot: A hybrid learning framework to uncover bot install fraud in mobile advertising. In *CIKM'20*, pages 2901–2908, 2020.

[58] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen. Graph contrastive learning with augmentations. *NIPS'20*, 33, 2020.

[59] Y. Zhang, Y. Fan, Y. Ye, L. Zhao, and C. Shi. Key player identification in underground forums over attributed heterogeneous information network embedding framework. In *CIKM'19*, 2019.

Bo Chen is a PhD candidate in the Department of Computer Science and Technology, Tsinghua University. He got his master's degree from the information school, Renmin University of China. His research interests include data integration and knowledge graph mining. He has published some related papers at the top conferences and journals such as TKDE, AAAI, IJCAI, and so on.



Jing Zhang is an associate professor at School of Information, Renmin University of China. Prior to that, She received her Ph.D. degree from the Department of Computer Science and Technology in Tsinghua University. Her current research interest falls in graph neural networks and knowledge graph reasoning. She has published more than 50 papers at the top conferences and journals in the area of data mining and artificial intelligence such as KDD, SIGIR, IJCAI, AAAI, TKDE, and so on.

Xiaokang Zhang is an undergraduate student in Information School, Renmin University of China. His research interests includes knowledge graph mining.

PLACE PHOTO HERE

Yuxiao Dong is a research scientist at Facebook AI, Seattle and was previously a senior researcher at Microsoft Research, Redmond. He received his Ph.D. in Computer Science from University of Notre Dame in 2017. His research focuses on data mining, representation learning, graph neural networks, knowledge graphs, and social & information networks, with an emphasis on developing machine learning models to addressing problems in Web-scale systems, such as Facebook Entity Graph, Microsoft Graph, and Microsoft Academic Graph (MAG).

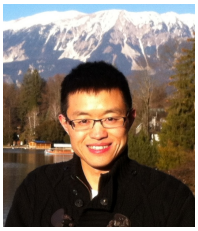
PLACE PHOTO HERE

Jian Song is a research engineer at Zhupu.AI. His mainly work includes author name disambiguation algorithm and paper data processing pipeline.

Peng Zhang is a senior engineer in the Department of Computer Science and Technology, Tsinghua University, and also a Ph.D. of Tsinghua University Innovation Leadership Project. He focused in text mining, knowledge graph construction and application.

Kaibo Xu received his Bachelor degree (1998) in Computer Science from Beijing University of Chemical Technology and his Master (2005) and PhD (2010) in Computer Science from the University of the West of Scotland. He worked as a Teaching Assistant (1998-2004), Lecturer (2004-2009), Associate Professor (2009-2017) at Beijing Union University. He has supervised more than 20 master and doctoral students who are successful in their academic and industrial careers. As the principal investigator, he has received 7 governmental funds and 5 industrial funds with the total amount of 5M in the Chinese dollar. Dr. Kaibo Xu has also consulted extensively and been involved in many industrial projects. He worked as the Chief-Information-Officer (CIO) of Yunbai Clothing Retail Group, China (2016-2019). Currently, he is serving as the vice president and principal scientist of MiningLamp Tech. His research interests include graph mining, knowledge graph and knowledge reasoning.

Evgeny Kharlamov is a Senior Expert at the Bosch Centre for Artificial Intelligence and an Associate Professor at the University of Oslo. He received his PhD degree in 2011 from the Free University of Bozen-Bolzano in cooperation with INRIA Saclay and Teleocom ParisTech. He worked as a Senior Researcher (from 2013 till 2018) at the University of Oxford and was a visiting researcher at the University of Edinburgh in 2011. His research interests are centered around Neuro-Symbolic AI methods that combine Knowledge Graphs and Machine Learning with applications in smart manufacturing. He has published more than 130 papers in major international journals and conferences.



Jie Tang is a Professor and the Associate Chair of the Department of Computer Science at Tsinghua University. He is a Fellow of the IEEE. His interests include artificial intelligence, data mining, social networks, and machine learning. He was honored with the SIGKDD Test-of-Time Award, the UK Royal Society-Newton Advanced Fellowship Award, NSFC for Distinguished Young Scholar, and KDD'18 Service Award.