FGAD: Self-boosted Knowledge Distillation for An Effective Federated Graph Anomaly Detection Framework

Jinyu Cai*
jinyucai@nus.edu.sg
Institute of Data Science, National
University of Singapore
Singapore

Yunhe Zhang*
zhangyhannie@163.com
School of Data Science, The Chinese
University of HongKong (Shenzhen)
China

Zhoumin Lu
walker.zhoumin.lu@gmail.com
School of Computer Science,
Northwest Polytechnical University
China

Wenzhong Guo
guowenzhong@fzu.edu.cn
College of Computer and Data
Science, Fuzhou University
China

See-Kiong Ng
seekiong@nus.edu.sg
Institute of Data Science, National
University of Singapore
Singapore

ABSTRACT

Graph anomaly detection (GAD) aims to identify anomalous graphs that significantly deviate from other ones, which has raised growing attention due to the broad existence and complexity of graph-structured data in many real-world scenarios. However, existing GAD methods usually execute with centralized training, which may lead to privacy leakage risk in some sensitive cases, thereby impeding collaboration among organizations seeking to collectively develop robust GAD models. Although federated learning offers a promising solution, the prevalent non-IID problems and high communication costs present significant challenges, particularly pronounced in collaborations with graph data distributed among different participants. To tackle these challenges, we propose an effective federated graph anomaly detection framework (FGAD). We first introduce an anomaly generator to perturb the normal graphs to be anomalous, and train a powerful anomaly detector by distinguishing generated anomalous graphs from normal ones. Then, we leverage a student model to distill knowledge from the trained anomaly detector (teacher model), which aims to maintain the personality of local models and alleviate the adverse impact of non-IID problems. Moreover, we design an effective collaborative learning mechanism that facilitates the personalization preservation of local models and significantly reduces communication costs among clients. Empirical results of the GAD tasks on non-IID graphs compared with state-of-the-art baselines demonstrate the superiority and efficiency of the proposed FGAD method.

*Both authors contributed equally to this research.

1 INTRODUCTION

Anomaly detection [3, 27] is a fundamental research problem in machine learning, and it has been extensively explored in various domains such as images [2, 15] and time-series data [1, 4, 20]. In the real world, graph-structured data is commonly available due to its exceptional ability to represent complicated relationship information among entities [43]. This is particularly evident in domains like social networks and medical applications. Consequently, graph anomaly detection (GAD) [5, 24], which aims to identify graphs that exhibit significant deviations from other normal graphs, has raised broad attention in recent years. With the advancement of graph neural networks (GNNs) [13, 37], GAD has made remarkable strides and demonstrated promising performance in detecting anomalies across many real-world scenarios with natural graph-structured data, e.g., social networks, molecules, and bioinformatics.

In realistic collaborative efforts among different companies and organizations, they attempt to share knowledge with each other in order to more accurately detect anomalies. However, existing GAD approaches [9, 23, 31, 40] typically involve a centralized model that requires all participants to provide their own data for training a global model, as shown in Figure 1(a). Although this centralized training simplifies coordination, it introduces a critical privacy

leakage risk. Graph data may encompass some sensitive information that the participant is not willing to share, e.g., the private relationship in social networks, which then hinders their collaborations. Consequently, an urgent imperative emerges to investigate approaches that facilitate collaboration between GAD models distributed to different participants while protecting their privacy.
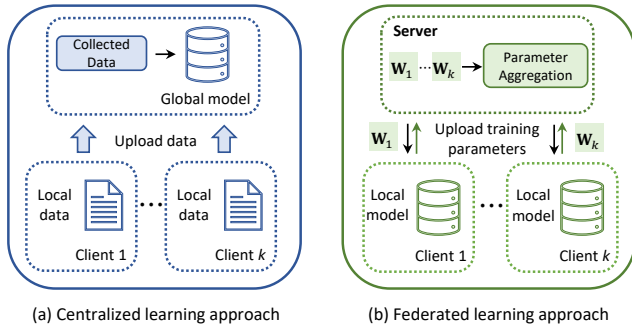

Figure 1: Overview of the centralized learning and federated learning frameworks.

As the emerging technique in machine learning, federated learning (FL), as shown in Figure 1(b), enables the collaboration between different participants with the consideration of privacy-preserving. The clients in FL only need to share their network parameters with the server rather than their local data, which prevents the leakage of sensitive information in participants. Classical FL methods, such as FedAvg [25] and FedProx [18], have become the paradigm of collaborative learning across various domains [14, 35]. To facilitate the collaborative training of GNN models for graph data across clients, federated graph learning (FGL) [8, 19, 41] has also been widely studied in recent years. FGL methods [30, 36] integrate GNNs with FL methods to collaboratively learn representations for complicated graph data distributed in various clients, and have demonstrated superiority in graph classification tasks. Therefore, an intuitive approach to address the above issue is to integrate the existing advancements in FL and FGL with general anomaly detection techniques, e.g., deep one-class classification (DeepSVDD) [29].

However, this solution may encounter the following challenges:

(1) The graph data distributed in various clients often exhibits significant heterogeneity and non-IID property [10, 36], e.g., containing different graph structures or feature dimensions. These factors place a higher demand on maintaining the validity of the local models for their own data, e.g., personalization.

(2) It is difficult to learn a universal hypersphere as the decision boundary for highly heterogeneous graph data under the federated learning setting. Besides, such non-IID graphs across clients hardly conform to the assumption in DeepSVDD that their latent distribution could follow a universal hypersphere.

(3) Existing collaborative learning mechanisms, e.g., FedAvg [25], require transmitting all network parameters of each client in a single communication round, which brings substantial communication costs in applications.

Those challenges naturally lead to a research question: ***Can we design an FL-based GAD framework that facilitates more effective collaboration and achieves more accurate detection?***

In this paper, we propose an effective federated graph anomaly detection (FGAD) framework, as shown in Figure 2, to answer this research question. To improve the anomaly detection capability in the local model, we introduce an anomaly generator that perturbs normal graphs to be anomalous, and train a classifier to identify anomalies from normal graphs. The generated anomalous graphs are encouraged to be diverse and resemble normal ones through iterations, so that more robust decision boundaries can be learned in a self-boosted manner. To alleviate the adverse impact of non-IID problems, we propose to preserve the personalization of each client by leveraging knowledge distillation. Specifically, we introduce a student model to distill the knowledge from the trained classifier (teacher model). The student model only takes the normal graphs as the input, with the aim of aligning its predicted distributions with that of the teacher model. Moreover, we further design an effective collaborative learning mechanism. We let the student and teacher models share the same backbone network to streamline the capacity of local models. Besides, we engage only the parameters of the student head rather than the entire model in collaborative learning, which allows the teacher model to preserve the personalization of a client. In this way, we not only alleviate the adverse impact of non-IID property, but also reduce the communication costs between clients and server during collaborative learning. The contributions of this paper are summarized as follows:

- We investigate the challenging anomaly detection issue on non-IID graphs distributed across various clients, and propose an effective federated graph anomaly detection (FGAD) framework.
- We introduce a self-boosted distillation module, which not only promotes the detecting capability by identifying self-generated anomalies, but also maintains the personalization of local models from knowledge distillation to alleviate non-IID problems.
- We propose an effective collaborative learning mechanism that streamlines the capacity of local models and reduces communication costs with the server.
- We establish a comprehensive set of baselines for federated graph anomaly detection. Extensive experiments also validate the effectiveness of the proposed FGAD method.

2 RELATED WORKS

2.1 Graph Anomaly Detection

Graph anomaly detection (GAD) [24] refers to detecting abnormal graphs that significantly differ from other normal ones, which have received growing attention in recent years owing to the ubiquitous prevalence of graph-structured data in real-world scenarios, such as social networks [22]. There are many works that advance the research on GAD. For instance, Zhao et al. [42] investigated graph-level anomaly detection issues by integrating graph isomorphism network (GIN) [37] with deep one-class classification (DeepSVDD) [29]. Qiu et al. [28] leveraged neural transformation learning to develop a more robust GAD model to overcome the performance flip issue. Ma et al. [23] utilized knowledge distillation to capture more comprehensive normal patterns from the global and local views for detecting graph anomalies.
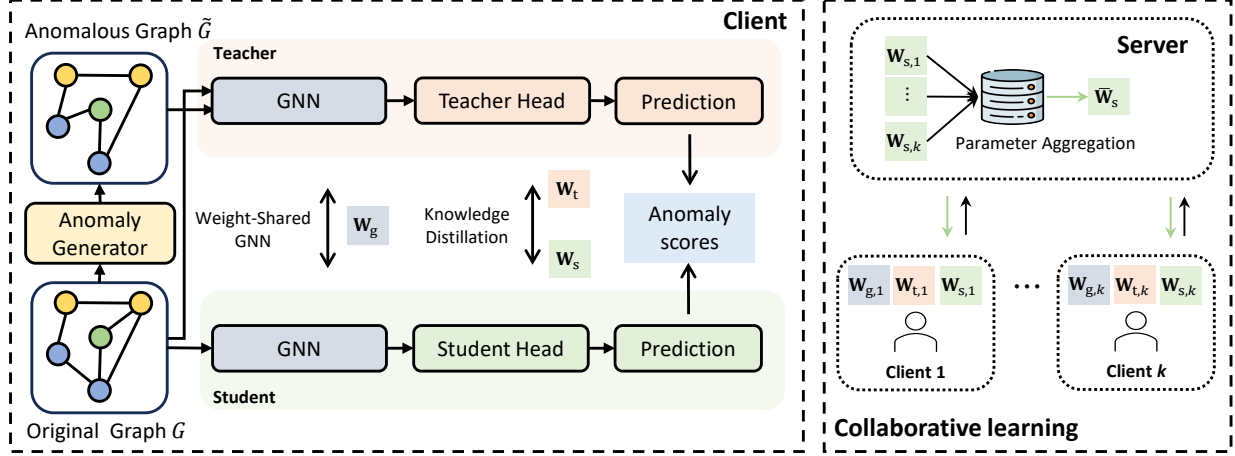
Figure 2: Overview of the FGAD framework. Note that the teacher model utilizes both normal and generated anomalous graphs for training an anomaly detector, while the student model only inputs normal graphs for the distillation of normal patterns.

Although these GAD methods have achieved remarkable success, they primarily rely on centralized training paradigms. Nevertheless, in real-world collaborative scenarios, the graph data is often distributed across various clients, which necessitates the transmission of local graph data to a central server during practical collaborations. Unfortunately, this process can potentially expose sensitive information and pose severe privacy risks. Additionally, the inherent non-IID property in the graph data distributed across diverse clients presents yet another formidable challenge. Consequently, the pursuit of effective solutions to address these challenges remains an open research problem.

2.2 Federated Graph Learning

Federated learning (FL) approaches [10, 17, 39], such as FedAvg [25], FedProx [18], provide a promising solution for collaboratively training models with data distributed in different clients, while preserving their privacy. In FL, clients only share their network parameters rather than data with the central server, which mitigates the privacy leakage risk and enables clients to share and leverage knowledge from others. As an emerging technique, FL has not only made remarkable advancements in image [6, 16, 38] and time series data [21, 33], but also raised increasing attention to graph data [34, 41], where collaborative efforts are significantly more challenging due to the complex structural information and heterogeneous characteristic of graphs compared to other data types.

Federated graph learning (FGL) [41] aims to facilitate the collaboration of GNNs distributed in multiple remote clients to meet the requirement of handling complicated non-IID graph data that widely exist in many real-world scenarios, e.g., social networks, medical, and biological data. For example, Xie et al. [36] studied the federated learning issue on non-IID graphs by integrating the clustered federated learning with graph isomorphism network (GIN), which achieves effective collaborations for distributed GINs. Tan et al. [30] designed a structural knowledge-sharing mechanism to facilitate the federated graph learning process. However, existing FGL methods have primarily been validated for graph classification tasks,

and their effectiveness in addressing the intricate unsupervised task of graph anomaly detection remains an area of ongoing exploration. While it is possible to extend these FL/FGL [18, 25, 30, 36] methods to address GAD tasks by integrating them with classical solutions like DeepSVDD [24, 29], it is imperative to acknowledge some significant challenges, e.g., the adverse impact of the non-IID problem across different clients and the communication costs of transmitting complex GNN model parameters during collaborative learning.

3 METHODOLOGY

3.1 Preliminary and Problem Formulation

Notation: Let $D = \{G_1, \ldots, G_N\}$ denotes a graph dataset which consists of N graphs, and each graph $G_i = \{V_i, E_i\}$ in the graph set comprises a node set V_i and edge set E_i. Typically, assume the number of nodes in a graph G_i is $n_i = |V_i|$, an adjacency matrix $\mathbf{A}_i \in \{0, 1\}^{n_i \times n_i}$ is used to represent the topology of graph G_i. Besides, let $\mathbf{x}_v \in \mathbb{R}^d$ denotes the attribute vector for node $v \in V_i$, $\mathbf{X}_i \in \mathbb{R}^{n_i \times d}$ is used to represent the attribute matrix of graph G_i.

Graph Neural Networks: Graph neural networks (GNNs), which iteratively learn representations with neighborhood aggregation and message propagation, is a widely used paradigm of learning representation for graph-structured data in many downstream tasks. In this paper, we leverage the graph isomorphism network (GIN) [37], a widely used GNN backbone, to learn graph representation for anomaly detection tasks. Generally, in each layer of a GIN, the node representation is updated by aggregating its neighborhood information. For instance, in the k-th layer of GIN, the learned aggregated features $\mathbf{a}_v^{(k)}$ for node v can be formulated as:

$$\mathbf{a}_v^{(k)} = \text{AGGREGATE}(\{\mathbf{h}^{(k-1)}(u), u \in \tilde{\mathcal{N}}(v)\}), \qquad (1)$$

where $\text{AGGREGATE}(\cdot)$ indicates the aggregation function, and $\tilde{\mathcal{N}}(v)$ represents the neighbor node set of node v. Then, the node feature $\mathbf{h}_v^{(k)}$ of node v in the k-th layer is obtained by combing the node feature learned in the $(k-1)$-th layer with the aggregated

feature, i.e.:

$$\mathbf{h}_v^{(k)} = \sigma(\text{COMBINE}(\mathbf{h}_v^{(k-1)}, \mathbf{a}_v^{(k)})), \tag{2}$$

where $\sigma(\cdot)$ denotes the activation function, e.g., LeakyReLU. Particularly, the initial feature $\mathbf{h}_v^{(0)}$ for node v is set as $\mathbf{h}_v^{(0)} = \mathbf{x}_v$. Consequently, we can obtain the representation for a graph G based on the learned features of all nodes within G as follows:

$$\mathbf{h}_G = \mathcal{R}(\text{CONCAT}(\mathbf{h}_v^{(k)}, k \in \{1, \ldots, K\}), v \in G), \tag{3}$$

where K is the number of GIN layers, and $\text{CONCAT}(\cdot)$ denotes the concatenate operation that stacks the graph representation learned across all K layers. $\mathcal{R}(\cdot)$ denotes the readout function that obtains the graph-level representation by aggregating the node features within a graph, and we choose sum-readout in this paper. Note that for convenience, we use $\text{GIN}(\cdot)$ to simply represent a GIN model containing the above three operations, in the following sections.

Problem Formulation: The objective of the GAD under the FL setup is to facilitate collaboration among clients, which allows each participant to enhance their GAD models by leveraging knowledge from others without exposing private data. Given C clients, the collective graph dataset is denoted as $D = \{D_1, \ldots, D_C\}$, where each client possesses its own graph set D_c. A prevalent paradigm in GAD [24] is that all graphs within the client, i.e., $\forall G_i \in D_c$, are deemed as "normal". The model is trained to capture this normality so that the trained model can distinguish an "anomalous" graph $\tilde{G}$ deviates significantly from the distribution of D_c by some predefined assumptions, e.g., the hypersphere decision boundary in DeepSVDD [29]. Conversely, in this paper, we attempt to develop a classifier-based anomaly detector, which can adaptively learn decision boundaries rather than relying on the strong assumption of the shape of the latent distribution. This can be regarded as solving the following problem:

$$\underset{\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(C)}}{\text{minimize}} \quad \frac{1}{C} \sum_{c=1}^{C} \frac{|D_c|}{|D|} (\ell_c(y, f_{\mathbf{w}^{(c)}}(G)) + \ell_c(\tilde{y}, f_{\mathbf{w}^{(c)}}(\tilde{G}))), \tag{4}$$

where $|D|$ and $|D_c|$ denote the total number of graphs and that of in c-th client. $\{G; y\}$ represents the normal graph labeled with $y = 1$, and $\{\tilde{G}, \tilde{y}\}$ represents the anomalous graph labeled with $y = 0$. $\ell_c(\cdot)$ denotes the local loss function of c-th client, e.g., binary cross-entropy loss. $f_{\mathbf{w}^{(c)}}(\cdot)$ is the GIN-based neural network of c-th client, which is parameterized by $\mathbf{w}^{(c)}$. However, tackling this problem presents the following challenges:

1) GAD is generally an unsupervised task that only normal graph $\{G; y\}$ is accessible. Thus, how to produce high-quality anomalous graph $\{\tilde{G}; \tilde{y}\}$ for training each local anomaly detector?

2) In the context of FL-based GAD, how to alleviate the adverse impact of the non-IID property that is prevalent in the graph data across clients?

3) Transmitting all network parameters following conventional FL methods may limit the scalability given the complexity of GIN. Therefore, how to reduce communication costs in collaborative learning while maintaining the validity of local models?

3.2 Self-boosted Graph Knowledge Distillation

The first challenge raises the demand to produce anomalous graphs without using any supervised information. To this end, we propose a

graph anomaly generator denoted as $\mathcal{G}_{\mathbf{w}_a}(\cdot)$ to generate anomalous graphs by perturbing the graph structure of normal graph G. For each client, we aim to generate an anomalous graph set $\tilde{D}_c = \{\mathbf{X}_c, \tilde{\mathbf{A}}_c\}$ in an unsupervised manner by feeding with normal graph set D_c. To ensure diversity in the generated anomalous graphs, we leverage variational graph auto-encoder (VGAE) [12] to build the anomaly generator. Specifically, we first learn a latent Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\sigma}_c^2)$, which can be determined as follows:

$$\boldsymbol{\mu}_c = \text{GIN}_{\boldsymbol{\mu}}(\mathbf{X}_c, \mathbf{A}_c), \boldsymbol{\sigma}_c = \text{GIN}_{\boldsymbol{\sigma}}(\mathbf{X}_c, \mathbf{A}_c), \tag{5}$$

where $\text{GIN}_{\boldsymbol{\mu}}(\cdot)$ and $\text{GIN}_{\boldsymbol{\sigma}}(\cdot)$ denotes two distinct GINs in anomaly generator, and $\boldsymbol{\mu}_c$ and $\boldsymbol{\sigma}_c$ explicitly parameterize the following inference model:

$$q(\tilde{\mathbf{Z}}_c | \mathbf{X}_c, \mathbf{A}_c) = \prod_{i=1}^{|D_c|} q(\mathbf{Z}_c^{(i)} | \mathbf{X}_c, \mathbf{A}_c), \tag{6}$$

where $q(\tilde{\mathbf{Z}}_c^{(i)} | \mathbf{X}_c, \mathbf{A}_c) = \mathcal{N}(\tilde{\mathbf{Z}}_c^{(i)} | \boldsymbol{\mu}_c^{(i)}, \text{diag}(\boldsymbol{\sigma}_c^{(i)}))$, and it allows us to sample from a wide range in the latent space thereby facilitating the diverse anomalous graph generation. Here, we employ the reparametrization trick [11] to address the obstacle of gradient propagation in the sample operation. Consequently, the generated adjacency matrix can be calculated by:

$$\tilde{\mathbf{A}}_c = \mathcal{T}(\tilde{\mathbf{Z}}_c^\top \tilde{\mathbf{Z}}_c), \quad \tilde{\mathbf{Z}}_c = \boldsymbol{\mu}_c + \epsilon \boldsymbol{\sigma}_c, \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{1}), \tag{7}$$

where $\mathcal{T} : \mathbb{R} \to [0, 1]$ represents the element-wise transformation operations such as Sigmoid$(\cdot)$, and ϵ represents a random Gaussian noise that follows the standard normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{1})$.

Intuitively, allowing the generated graphs to closely resemble normal graphs while remaining as anomalies is beneficial in training a robust and powerful anomaly detector, as it forces the model to distinguish those subtle deviations from the normal patterns. Therefore, we propose to optimize the anomaly generator by minimizing the following objective:

$$\ell_g^c(\mathbf{A}_c, \tilde{\mathbf{A}}_c) = -\sum_{i,j} (\mathbf{A}_c^{ij} \log(\tilde{\mathbf{A}}_c^{ij}) + (1 - \mathbf{A}_c^{ij}) \log(1 - \tilde{\mathbf{A}}_c^{ij})), \tag{8}$$

where ℓ_g^c denotes the binary-cross entropy loss function. Subsequently, we can train an anomaly detector with the normal and generated anomalous graph sets for the local client as follows:

$$\ell_{\text{ad}}^c = l_{\text{ce}}(y_c, \text{Proj}(f_{\mathbf{w}_g}(\mathbf{X}_c, \mathbf{A}_c))) + l_{\text{ce}}(\tilde{y}_c, \text{Proj}(f_{\mathbf{w}_g}(\mathbf{X}_c, \tilde{\mathbf{A}}_c))), \tag{9}$$

where $\tilde{\mathbf{A}}_c = \mathcal{G}_{\mathbf{w}_a}(\mathbf{X}_c, \mathbf{A}_c)$, $l_{\text{ce}}(\cdot)$ is the cross-entropy loss, and $f_{\mathbf{w}_g}(\cdot)$ denotes the GIN backbone that learns graph representation by feeding with graph data. $\text{Proj}(\cdot)$ is the MLP-based projection head that maps the graph representation learned from $f_{\mathbf{w}_g}(\cdot)$ into the predicted logits. Note that we simply set the label of the normal graph $y_c = 1$, and the generated anomalous graph as $\tilde{y}_c = 0$.

Hence, we can train an anomaly detector in an unsupervised manner by minimizing the following objective function:

$$\ell_{\text{pt}} = \frac{1}{C} \sum_{c=1}^{C} \frac{|D_c|}{|D|} (\ell_{\text{ad}}^c + \ell_g^c), \tag{10}$$

where ℓ_g^c attempts to generate anomalous graphs that closely resemble normal ones, while ℓ_{ad}^c aims to identify those generated anomalous graphs. Therefore, we produce diverse anomalous graphs for learning a powerful anomaly detector in such a self-boosted style, and the two objectives mutually improve each other during training.

However, in the context of federated learning, the graph data across different clients is often heterogeneous and exhibits non-IID property. Such characteristics can potentially affect the anomaly detection performance of local models, i.e., the second challenge. To alleviate the adverse impact of the non-IID problem, we propose a graph knowledge distillation framework, which is designed to preserve the personalization of the local model during collaborative learning. Specifically, we regard the previously pre-trained anomaly detector as the "teacher" model, and introduce a "student" model that aims to distill the knowledge from the teacher model and achieve collaboration between clients.

The network architecture of the student model is similar to the teacher model, which consists of a GIN backbone and a projection head. Since the purpose of the student model is to mimic the predictions of the teacher model for normal data, only normal graphs are considered in the knowledge distillation. The predicted logits of the teacher and student models are computed as follows:

$$\mathbf{Q}_{c,\text{t}} = \text{Proj}_{\text{t}|\mathbf{w}_t}(f_{\mathbf{w}_g}(\mathbf{X}_c, \mathbf{A}_c)), \quad \mathbf{Q}_{c,\text{s}} = \text{Proj}_{\text{s}|\mathbf{w}_s}(f_{\mathbf{w}_{g'}}(\mathbf{X}_c, \mathbf{A}_c)), \quad (11)$$

where $f_{\mathbf{w}_g}(\cdot)$, $\text{Proj}_{\text{t}|\mathbf{w}_t}(\cdot)$ and $f_{\mathbf{w}_{g'}}(\cdot)$, $\text{Proj}_{\text{s}|\mathbf{w}_s}(\cdot)$ are the backbone networks and projection heads of teacher and student models respectively. Note that $\text{Proj}_{\text{t}|\mathbf{w}_t}(\cdot)$ is actually the same as the projection head $\text{Proj}(\cdot)$ in Eq. (9). Subsequently, the student model distills the knowledge from the teacher model by matching its predicted logits with those of the teacher model, described as follows:

$$\ell_{\text{kd}}^c = \frac{1}{|D_c|} \sum_{i \in D_c} KL(\text{softmax}(\mathbf{Q}_{c,\text{t}}^{(i)}/\tau), \text{softmax}(\mathbf{Q}_{c,\text{s}}^{(i)}/\tau)), \quad (12)$$

where $KL(\cdot, \cdot)$ denotes the Kullback-Leibler divergence, which is applied to measure the discrepancy between the distribution of the predicted logits from teacher and student models. softmax$(\cdot)$ is the softmax function, i.e., softmax$(q_i/\tau) = \frac{\exp(q_i/\tau)}{\sum_j \exp(q_j/\tau)}$, and τ is the temperature factor that controls the smoothness of the distillation.

3.3 Parameter-efficient Collaborative Learning

Based on the design of the self-boosted graph knowledge distillation module, the objective function of all clients is defined as follows:

$$\mathcal{L}_{\text{total}} = \frac{1}{C} \sum_{c=1}^{C} \frac{|D_c|}{|D|}(\ell_{\text{ad}}^c + \lambda\ell_{\text{g}}^c + \gamma\ell_{\text{kd}}^c), \quad (13)$$

where λ and γ are the two trade-off parameters. In federated learning, let $\mathbf{W}^{(c)} = \{\mathbf{w}_a^{(c)}, \mathbf{w}_g^{(c)}, \mathbf{w}_{g'}^{(c)}, \mathbf{w}_t^{(c)}, \mathbf{w}_s^{(c)}\}$ denotes the parameter set of the c-th client, the conventional solution achieves collaboration by uploading the network parameters to the server and then distribute the aggregated network parameters to each client. However, this solution presents several problems. First, the high parameter complexity of a GIN-based backbone can limit the scalability of the model during the parameter aggregation process. Second, the transmission of all network parameters may introduce non-IID problems, and affect the performance of local models trained on different graph data across clients.

To address these issues, we propose an effective collaborative learning mechanism in this paper, which is described in Figure 2. Specifically, We let the teacher and student models share the same

GIN backbone for learning graph representation, i.e.,

$$\mathbf{Z}_c = f_{\mathbf{w}_g}(\mathbf{X}_c, \mathbf{A}_c) = f_{\mathbf{w}_{g'}}(\mathbf{X}_c, \mathbf{A}_c), \quad (14)$$

where $\mathbf{Z}_c$ denotes the learned graph representation that is shared as the input to the projection heads of teacher and student. This operation not only reduces the complexity of the local model, but also simplifies the knowledge distillation of the student model. Then we only upload the parameter set $\mathbf{w}_s^{(c)}$ of the student head for collaboration instead of uploading all the network parameters, i.e., the parameter aggregation in the server is formalized as follows:

$$\bar{\mathbf{w}}_s = \sum_{c=1}^{C} \frac{|D_c|}{|D|}\mathbf{w}_s^{(c)}, \quad (15)$$

where $\bar{\mathbf{w}}_s$ denotes the aggregated parameters in the server. The proposed collaborative learning mechanism not only streamlines the capacity of local models, but also significantly reduces the communication costs, which addresses the third challenge. To facilitate the understanding of the proposed FGAD method, we summarize its detailed training process in Algorithm 1. The collaboration between clients via the student model is performed in the following two steps:

- Each client performs graph knowledge distillation independently, updating its network parameters, and uploads the network parameters of the student head to the server.
- The server then aggregates the network parameters following Eq. 15, and distributes the aggregated network parameters to each client.

Algorithm 1 Training process of the proposed FGAD

Input: Graph set $D = \{D_c\}_{c=1}^{C}$, number of clients C, number of GNN layers K, learning rate α, total epochs $\mathcal{T}$.
Output: The overall graph anomaly detection performance.
1: Initialize the parameter sets $\{\mathbf{W}^{(c)}\}_{c=1}^{C}$ for each local model;
2: Pretrain the local model in each client with Eq. (10);
3: **while** not converge **do**
4:     **for** $t = 1, 2, \ldots, \mathcal{T}$ **do**
5:         **for** $c = 1, \ldots, C$ **do**
6:             Generate anomalous graph set $\tilde{\mathbf{D}}$ with Eqs. (5), (6), (7);
7:             Compute loss items $\ell_{\text{ad}}^c, \ell_{\text{g}}^c, \ell_{\text{kd}}^c$ with Eq. (8), (9), (12);
8:         **end for**
9:     Back-propagation and update each local model via minimizing Eq. (13);
10:     **end for**
11:     Upload the parameter sets $\{\mathbf{w}_s^{(c)}\}_{c=1}^{C}$ of student model in each client to the server;
12:     Compute aggregated network parameters $\bar{\mathbf{w}}_s$ with collaborative learning following Eq. (15);
13:     Distribute parameter set $\bar{\mathbf{w}}_s$ to the local model of each client;
14: **end while**
15: Evaluate the anomaly detection performance in each client and aggregate their results;
16: **return** The overall graph anomaly detection performance.

---

4 EXPERIMENT

4.1 Experimental Setup

Datasets. We evaluate the performance of FL-based graph anomaly detection on non-IID graphs through two distinct experimental setups: (1) single-dataset and (2) multi-dataset scenarios.

- **Single-dataset:** we distribute a single dataset across multiple clients, each of which possesses a unique subset of the dataset. This setup allows us to assess the effectiveness when clients collaborate on a shared dataset. We employ three social network datasets including IMDB-BINARY, COLLAB, and IMDB-MULTI to conduct this experiment.
- **Multi-dataset:** we broaden our evaluation by considering various datasets distributed in multiple clients and each of them holds a specific dataset. We consider not only social network data (SO-CIALNET) but also expand to include molecular (MOLECULES), biochemical (BIOCHEM), and mix data types (MIX). This allows us to thoroughly assess FL-based graph anomaly detection across a spectrum of data types and collaboration scenarios.

The information and construction details of each dataset are illustrated in Appendix A.1.

Baseline Methods. We compare the proposed FGAD method with several state-of-the-art baseline methods. We include two federated learning methods: FedAvg [25] and FedProx [18], as well as two federated graph learning methods: GCFL [36] and FedStar [30]. Note that in order to adapt these baseline methods to the graph anomaly detection task, we integrate them with DeepSVDD [29] to construct an end-to-end graph anomaly detection model. Besides, we regard the self-training strategy without the FL setting as one of the baselines. To ensure a fair comparison with FGAD, we employ the same GIN network structure as FGAD in all baseline methods.

Implementation Details. We use GIN [37] as the graph representation learning backbone for FGAD and all baselines. The number of GIN layer K is set to 3, and the dimensions of the hidden layer of GIN and projection head of student and teacher models are all set to 64. We use Adam [11] as the optimizer and fixed the learning rate $\alpha = 0.001$. For all datasets, we first pretrain the anomaly generator and teacher model for 10 epochs, and then jointly train with knowledge distillation and collaborative learning for 200 epochs. For more training details, please refer to Appendix A.2.

Evaluation Metrics: We use Area Under the Curve (AUC) and Area Under the Precision-Recall Curve (AUPRC) as the evaluation metrics in the experiment. Each method is executed 10 times to report their means and standard deviations.

4.2 Experimental Results

In this section, we conduct comprehensive experiments including two types of non-IID graph scenarios, i.e., the single-dataset and multi-dataset distributed in multiple clients, to validate the effectiveness of the proposed method. Table 1 and Table 2 show the experimental results of FGAD and several state-of-the-art baselines, from which we can have the following observations.

- **Comparison:** In the single-dataset experiment, FGAD demonstrates a remarkable advantage over all baseline methods. For

instance, in the IMDB-BINARY dataset, FGAD achieves significant performance improvement, exceeding Self-train by 23.39% in AUC and 19.17% in AUPRC. It also significantly surpasses classical FedAvg and FedProx. Furthermore, FGAD outperforms the state-of-the-art baselines GCFL and FedStar by a substantial 7.99% and 10.21% in AUC, respectively. Similar trends are evident across other benchmarks, demonstrating the effectiveness of FGAD. In the multi-dataset experiment, the GAD task is more challenging as the non-IID problem in it is more severe compared to the single-dataset scenario. Nevertheless, FGAD still exhibits outstanding performance compared to other baselines. For example, on MOLECULES, FGAD outperforms the runner-up FedStar by 6% in AUC and 19.46% in AUPRC. Besides, it achieves more than a 10.00% performance improvement compared to other baseline methods. More importantly, we can observe from Table 1 that FGAD significantly reduces communication costs during collaborative learning compared to other baseline methods.

- **Discussion:** The Self-train strategy discards collaborative training and fails to leverage the knowledge from other clients to learn more robust local GAD models. FedAvg and FedProx require the transmission of all network parameters of the local models, which introduces severe non-IID problems in collaborative learning. Consequently, these three aforementioned baselines yield suboptimal performance in most cases. Although GCFL incorporates a specific design to alleviate non-IID challenges, such as utilizing clustered FL for collaborative learning, it still necessitates the transmission of all network parameters and does not effectively address non-IID problems, as validated by the experimental results. On the other hand, FedStar achieves runner-up performance in most cases, which may primarily be attributed to the introduced structural embedding that helps to preserve the personalization of local models. Compared with the baseline methods, FGAD considers enhancing the detecting capability of local models in a self-boosted manner, and introduces an effective collaborative learning mechanism by leveraging knowledge distillation. This allows FGAD to learn more powerful local GAD models, mitigate the adverse effects of non-IID problems, and reduce communication costs among clients.

4.3 Embedding Visualization

We employ t-SNE [32] to visualize the learned embedding for intuitive comparison. Figure 3 shows the embedding visualization for AIDS, one of the constituents of MOLECULES. We include results from FedAvg, GCFL, and FedStar for a comprehensive analysis. It's evident that the learned embeddings by FedAvg and GCFL exhibit poor discriminative properties, with both normal and anomalous graphs appearing entangled in the latent space. Although the visualization result of FedStar shows some separation between normal and anomalous graphs, it remains blurred decision boundaries. Conversely, the learned embeddings of FGAD are clearly more discriminative compared to the other baseline methods. The visualization of FGAD reveals distinct boundaries between the embeddings of normal and anomalous graphs, supporting its effectiveness.

Table 1: Anomaly detection performance (mean(%) ± std(%)) under the single-dataset setting. Note that the best performance is marked in Bold, and the last column shows the number of transmitted parameters in collaborative learning.

Methods	IMDB-BINARY		COLLAB		IMDB-MULTI		# Parameters
	AUC	AUPRC	AUC	AUPRC	AUC	AUPRC	
Self-train	41.58±1.34	47.43±1.39	46.96±1.80	30.87±0.62	52.39±1.31	32.74±0.60	N/A
FedAvg [25]	40.96±3.44	48.24±2.41	49.60±0.45	30.69±0.50	49.11±1.46	36.13±1.54	5,370,880
FedProx [18]	39.62±2.36	46.74±1.24	49.56±0.50	31.40±0.50	52.16±1.75	36.13±1.54	5,370,880
GCFL [36]	56.98±5.56	59.68±3.37	48.93±1.02	30.84±0.36	49.44±2.95	34.87±0.68	10,741,760
FedStar [30]	54.76±1.28	56.49±0.86	51.89±0.33	36.89±0.43	58.28±0.53	39.97±1.22	416,000
FGAD	**64.97±0.52**	**66.60±1.12**	**55.08±1.85**	**66.67±0.00**	**60.51±1.18**	**66.82±0.14**	21,130

Table 2: Anomaly detection performance (mean(%) ± std(%)) under the multi-dataset setting. Note that the best performance is marked in Bold.

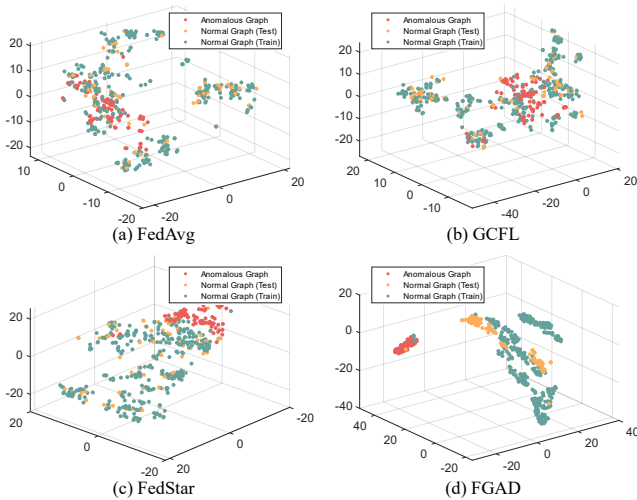| Methods	MOLECULES		BIOCHEM		SOCIALNET		MIX	
	AUC	AUPRC	AUC	AUPRC	AUC	AUPRC	AUC	AUPRC
Self-train	61.26±2.91	61.31±1.91	54.54±0.99	52.29±0.40	50.31±1.55	39.96±1.58	51.94±0.42	47.65±0.64
FedAvg [25]	54.41±3.21	55.55±3.23	40.88±1.36	51.63±1.13	48.21±1.02	38.29±1.29	47.96±0.61	44.89±0.68
FedProx [18]	57.93±2.14	58.72±2.25	46.04±0.49	51.57±0.80	47.26±0.10	37.23±0.92	46.79±0.63	44.19±0.29
GCFL [36]	45.67±1.33	51.96±0.79	41.49±0.30	52.23±0.65	47.59±0.95	37.53±0.93	49.58±0.50	45.37±0.69
FedStar [30]	56.15±0.92	59.73±1.21	47.80±0.48	56.48±0.19	53.79±2.03	36.40±1.11	50.53±1.11	45.83±0.41
FGAD	**62.15±0.69**	**79.19±0.49**	**58.09±0.85**	**59.04±0.54**	**54.86±0.29**	**56.88±0.98**	**58.14±0.36**	**52.03±0.63**

Figure 3: Embedding visualization of the proposed FGAD compared with several baselines using t-SNE. Note that the data point marked in yellow, red, and green correspond to the normal graph (test), anomalous graph, and normal graph (train), respectively.

Table 3: Ablation study results (mean(%) ± std(%)) of FGAD and its three variants.

Methods	IMDB-MULTI		MOLECULES	
	AUC	AUPRC	AUC	AUPRC
FGAD_v1	56.67±1.72	64.91±1.85	57.98±2.78	75.80±0.09
FGAD_v2	56.69±1.22	65.98±0.90	59.41±2.22	77.32±1.02
FGAD_v3	55.23±3.54	61.02±2.68	55.58±4.56	66.73±0.80
FGAD	**60.51±1.18**	**66.82±0.14**	**62.15±0.69**	**79.19±0.49**

4.4 Ablation Study

To validate the effectiveness of each component in the proposed FGAD method, we derive three variants from FGAD and perform a systematic evaluation. Specifically, we illustrate the construction details of the three variants as follows:

- **FGAD_v1:** This variant only considers local training in each client, and abandons the collaborative learning between clients.
- **FGAD_v2:** This variant drops the proposed collaborative learning mechanism, and follows the parameter aggregation mechanism of the classical FedAvg method.
- **FGAD_v3:** This variant drops the knowledge distillation module, i.e., removes the student model and only takes the classifier (teacher model) in collaboration.
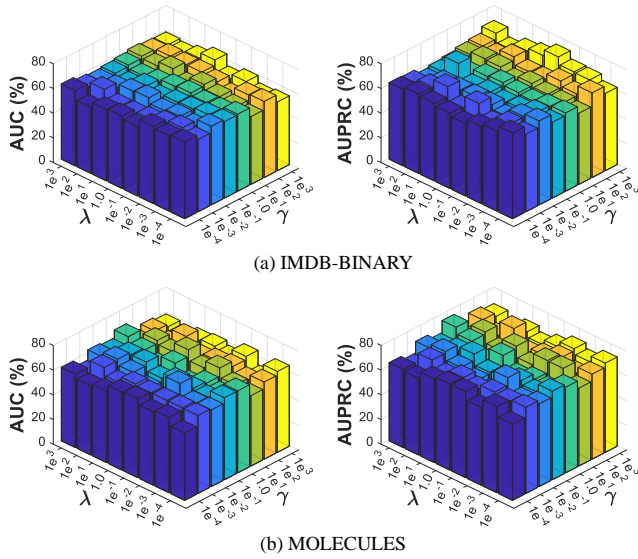
(a) IMDB-BINARY

(b) MOLECULES

Figure 4: Parameter analysis of λ and γ on IMDB-BINARY and MOLECULES. Note that the values of λ and γ range from $[1e^{-4}, \ldots, 1e^3]$.

Table 3 shows the experimental results of FGAD and its three variants on two datasets, yielding the following observations. FGAD_v1 demonstrates a performance decline compared to FGAD, which is primarily due to the fact that FGAD_v1 exclusively focuses on local training, neglecting collaboration with other clients. Consequently, it fails to leverage the comprehensive knowledge of other clients. Secondly, when we substitute the proposed collaborative learning mechanism with the classical FedAvg, there is also a noticeable performance decline. This can be attributed to the potential susceptibility of parameter transmission in FedAvg to the adverse effects of non-IID problems. Third, FGAD consistently outperforms FGAD_v3 by a significant margin. This observation reveals the crucial role of the self-boosted distillation module in maintaining the personalization of local models within each client, which effectively mitigates the non-IID problems. Overall, the ablation study results fully support the rationale and the effectiveness of each component proposed in FGAD.

4.5 Parameter Analysis

4.5.1 Impact of Hyper-Parameters λ and γ. The objective function of the proposed FGAD method contains two main hyper-parameters, i.e., λ and γ. In this section, we conduct an analysis of the impact of these two hyper-parameters on anomaly detection performance. Specifically, we vary the values of λ and γ within the range of $[1e^{-3}, \ldots, 1e^4]$ and present the experimental results on IMDB-BINARY and MOLECULES datasets in Figure 4. From the observations shown in the figure, we draw several conclusions. Firstly, FGAD tends to yield suboptimal performance when the values of λ and γ are set too low, e.g., $1e^{-4}$ and $1e^{-3}$. This emphasizes the significant role of both loss terms in the FGAD framework and suggests their effectiveness. Secondly, we can observe that excessively high values of λ and γ also have an adverse impact on performance,
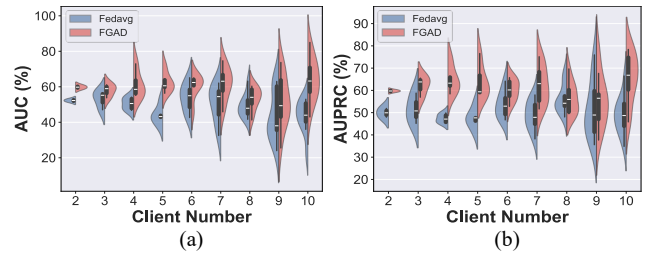

Figure 5: Average performance and distribution of variance between clients of FedAvg and FGAD. Note that the client number is set to $[2, \ldots, 10]$.
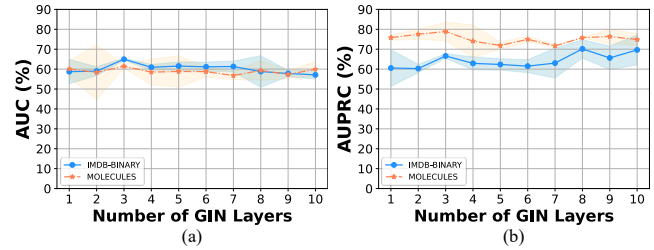

Figure 6: Average performance with standard deviation under different numbers of GIN layers on IMDB-BINARY and MOLECULES datasets. Note that the number of GIN layers is set to $[1, \ldots, 10]$.

because they may obscure the primary objective of optimizing the anomaly detector. Finally, it is worth noting that FGAD exhibits relatively stable performance both in AUC and AUPRC across a wide range of λ and γ values, demonstrating its robustness.

4.5.2 Impact of Client Numbers. The number of clients C is another hyper-parameter in the FGAD framework, and its impact on the performance is crucial for assessing the scalability of client numbers. Therefore, we vary the number of clients C within the range of $[2, \ldots, 10]$ and conduct the experiment. The results on IMDB-BINARY are reported in Figure 5. Note that we also include FedAvg as a baseline method for comparative analysis. It can be observed that FGAD consistently achieves remarkable performance improvement compared to FedAvg in all cases, and exhibits stability against changes in the number of clients. However, when the number of clients increases to certain large values, the average performance shows a certain degradation, and the performance variance between different clients becomes more significant both in FGAD and FedAvg. This is primarily due to the gradually increasing discrepancy between the graph data distributed across different clients, which causes more severe non-IID problems. Nevertheless, FGAD still exhibits relatively smaller performance fluctuations compared with FedAvg, which fully demonstrates the scalability of the proposed FGAD method.

4.5.3 Impact of GIN Layers. We delve into the impact of the number of GIN layers K on the anomaly detection performance within the proposed FGAD method. The parameter K plays a pivotal role in determining the extent to which the model explores

neighborhood information and the overall complexity of FGAD. We systematically analyze its impact by varying the K within the range of $[1, \ldots, 10]$ and conduct a series of experiments. Figure 6 reports the experimental results on the IMDB-BINARY and MOLECULES datasets, from which we have the following observations. First, a certain depth of GIN is beneficial to fully leverage the structural information of graph data for learning powerful GAD models, which could be verified from the observed performance improvement. Second, when the number of GIN layers continues to increase, the observed performance improvements become increasingly marginal or even exhibit slight diminishment. This trend indicates that a moderate number of GIN layers, e.g., 3, is sufficient to effectively leverage the neighborhood information within graphs. Third, we can observe from the overall experimental results that the performance remains relatively stable under the variation of K, which demonstrates the robustness of FGAD.

5   CONCLUSION

In this paper, we study a challenging GAD problem with non-IID graph data distributed across multiple clients, and propose an effective federated graph anomaly detection (FGAD) method to tackle this issue. To enhance the detecting capability of local models, we propose to train a classifier in a self-boosted manner by distinguishing the normal and anomalous graphs generated from an anomaly generator. Besides, to alleviate the adverse impact of non-IID problems among clients, we introduce a student model to distill knowledge from the classifier (teacher model), and engage only the student model in collaborative learning, so that the personalization of local models could be preserved. Moreover, we improve the collaborative learning mechanism that streamlines the capacity of local models and reduces the communication costs during collaborative learning. Comprehensive experiments under various data types and scenarios compared with state-of-the-art baselines demonstrate the superiority of the proposed FGAD method. We believe that this work would pave the way for subsequent studies on collaborative GAD under the FL setting in the future.

REFERENCES

[1] Ane Blázquez-García, Angel Conde, Usue Mori, and Jose A Lozano. 2021. A review on outlier/anomaly detection in time series data. *ACM Computing Surveys (CSUR)* 54, 3 (2021), 1–33.

[2] Jinyu Cai and Jicong Fan. 2022. Perturbation learning based anomaly detection. *Advances in Neural Information Processing Systems* 35 (2022).

[3] Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly detection: A survey. *ACM Computing Surveys (CSUR)* 41, 3 (2009), 1–58.

[4] Ailin Deng and Bryan Hooi. 2021. Graph neural network-based anomaly detection in multivariate time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 4027–4035.

[5] Kaize Ding, Qinghai Zhou, Hanghang Tong, and Huan Liu. 2021. Few-shot network anomaly detection via cross-network meta-learning. In *Proceedings of the Web Conference*. 2448–2456.

[6] Gokberk Elmas, Salman UH Dar, Yilmaz Korkmaz, Emir Ceyani, Burak Susam, Muzaffer Ozbey, Salman Avestimehr, and Tolga Çukur. 2022. Federated learning of generative image priors for MRI reconstruction. *IEEE Transactions on Medical Imaging* (2022).

[7] Matthias Fey and Jan Eric Lenssen. 2019. Fast graph representation learning with PyTorch Geometric. *arXiv preprint arXiv:1903.02428* (2019).

[8] Xingbo Fu, Binchi Zhang, Yushun Dong, Chen Chen, and Jundong Li. 2022. Federated graph machine learning: A survey of concepts, techniques, and applications. *ACM SIGKDD Explorations Newsletter* 24, 2 (2022), 32–47.

[9] Yuan Gao, Xiang Wang, Xiangnan He, Zhenguang Liu, Huamin Feng, and Yongdong Zhang. 2023. Alleviating structural distribution shift in graph anomaly

detection. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 357–365.

[10] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. 2021. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning* 14, 1–2 (2021), 1–210.

[11] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[12] Thomas N Kipf and Max Welling. 2016. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308* (2016).

[13] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *Proceedings of the International Conference on Learning Representations*.

[14] Chenglin Li, Di Niu, Bei Jiang, Xiao Zuo, and Jianming Yang. 2021. Meta-har: Federated representation learning for human activity recognition. In *Proceedings of the Web Conference*. 912–922.

[15] Chun-Liang Li, Kihyuk Sohn, Jinsung Yoon, and Tomas Pfister. 2021. Cutpaste: Self-supervised learning for anomaly detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9664–9674.

[16] Qinbin Li, Bingsheng He, and Dawn Song. 2021. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10713–10722.

[17] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. 2020. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine* 37, 3 (2020), 50–60.

[18] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems* 2 (2020), 429–450.

[19] Rui Liu, Pengwei Xing, Zichao Deng, Anran Li, Cuntai Guan, and Han Yu. 2022. Federated graph neural networks: Overview, techniques and challenges. *arXiv preprint arXiv:2202.07256* (2022).

[20] Shenghua Liu, Bin Zhou, Quan Ding, Bryan Hooi, Zhengbo Zhang, Huawei Shen, and Xueqi Cheng. 2022. Time series anomaly detection with adversarial reconstruction networks. *IEEE Transactions on Knowledge and Data Engineering* 35, 4 (2022), 4293–4306.

[21] Yi Liu, Sahil Garg, Jiangtian Nie, Yang Zhang, Zehui Xiong, Jiawen Kang, and M Shamim Hossain. 2020. Deep anomaly detection for time-series data in industrial IoT: A communication-efficient on-device federated learning approach. *IEEE Internet of Things Journal* 8, 8 (2020), 6348–6358.

[22] 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.

[23] Rongrong Ma, Guansong Pang, Ling Chen, and Anton van den Hengel. 2022. Deep graph-level anomaly detection by glocal knowledge distillation. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 704–714.

[24] Xiaoxiao Ma, Jia Wu, Shan Xue, Jian Yang, Chuan Zhou, Quan Z Sheng, Hui Xiong, and Leman Akoglu. 2021. A comprehensive survey on graph anomaly detection with deep learning. *IEEE Transactions on Knowledge and Data Engineering* (2021).

[25] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*. PMLR, 1273–1282.

[26] Christopher Morris, Nils M. Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. 2020. TUDataset: A collection of benchmark datasets for learning with graphs. In *Proceedings of the ICML Workshop on Graph Representation Learning and Beyond*. arXiv:2007.08663 www.graphlearning.io

[27] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton Van Den Hengel. 2021. Deep learning for anomaly detection: A review. *ACM Computing Surveys (CSUR)* 54, 2 (2021), 1–38.

[28] Chen Qiu, Marius Kloft, Stephan Mandt, and Maja Rudolph. 2022. Raising the Bar in Graph-level Anomaly Detection. In *Proceedings of the International Joint Conference on Artificial Intelligence*. 2196–2203. https://doi.org/10.24963/ijcai.2022/305

[29] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. 2018. Deep one-class classification. In *Proceedings of the International Conference on Machine Learning*. PMLR, 4393–4402.

[30] Yue Tan, Yixin Liu, Guodong Long, Jing Jiang, Qinghua Lu, and Chengqi Zhang. 2023. Federated learning on non-iid graphs via structural knowledge sharing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 9953–9961.

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

[32] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research* 9, 11 (2008), 2579–2605.

[33] Siqi Wang, Jiashu Li, Mian Lu, Zhao Zheng, Yuqiang Chen, and Bingsheng He. 2022. A system for time series feature extraction in federated learning. In *Proceedings of the ACM International Conference on Information & Knowledge Management*. 5024–5028.

[34] Zhen Wang, Weirui Kuang, Yuexiang Xie, Liuyi Yao, Yaliang Li, Bolin Ding, and Jingren Zhou. 2022. Federatedscope-gnn: Towards a unified, comprehensive and efficient package for federated graph learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4110–4120.

[35] Jinze Wu, Qi Liu, Zhenya Huang, Yuting Ning, Hao Wang, Enhong Chen, Jinfeng Yi, and Bowen Zhou. 2021. Hierarchical personalized federated learning for user modeling. In *Proceedings of the Web Conference*. 957–968.

[36] Han Xie, Jing Ma, Li Xiong, and Carl Yang. 2021. Federated graph classification over non-iid graphs. *Advances in Neural Information Processing Systems* 34 (2021), 18839–18852.

[37] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How powerful are graph neural networks?. In *Proceedings of the International Conference on Learning Representations*.

[38] Rui Yan, Liangqiong Qu, Qingyue Wei, Shih-Cheng Huang, Liyue Shen, Daniel Rubin, Lei Xing, and Yuyin Zhou. 2023. Label-efficient self-supervised federated learning for tackling data heterogeneity in medical imaging. *IEEE Transactions on Medical Imaging* (2023).

[39] Chengxu Yang, Qipeng Wang, Mengwei Xu, Zhenpeng Chen, Kaigui Bian, Yunxin Liu, and Xuanzhe Liu. 2021. Characterizing impacts of heterogeneity in federated learning upon large-scale smartphone data. In *Proceedings of the Web Conference*. 935–946.

[40] Ge Zhang, Zhenyu Yang, Jia Wu, Jian Yang, Shan Xue, Hao Peng, Jianlin Su, Chuan Zhou, Quan Z Sheng, Leman Akoglu, et al. 2022. Dual-discriminative graph neural network for imbalanced graph-level anomaly detection. *Advances in Neural Information Processing Systems* 35 (2022), 24144–24157.

[41] Huanding Zhang, Tao Shen, Fei Wu, Mingyang Yin, Hongxia Yang, and Chao Wu. 2021. Federated graph learning–a position paper. *arXiv preprint arXiv:2105.11099* (2021).

[42] Lingxiao Zhao and Leman Akoglu. 2021. On using classification datasets to evaluate graph outlier detection: Peculiar observations and new insights. *Big Data* (2021).

[43] Chenyi Zhuang and Qiang Ma. 2018. Dual graph convolutional networks for graph-based semi-supervised classification. In *Proceedings of the World Wide Web Conference*. 499–508.

A APPENDIX

The appendix includes the following content:

(1) Detailed description of graph benchmarks used in experiments.
(2) Detailed experimental settings.
(3) Theoretical and empirical complexity analysis.
(4) Parameter analysis of latent dimensions.
(5) Justification of the backbone sharing strategy.

A.1 Detailed Description of Graph Benchmarks

In this section, we supplement the detailed description for all the graph benchmarks used in our experiment, including the number of graphs, the average node numbers and edge numbers, and the classes. Table 4 summarizes the information on graph benchmarks used in the experiment. Specifically, in the single-dataset experiment, we use three social network benchmarks including IMDB-BINARY, COLLAB, and IMDB-MULTI. In the multi-dataset experiment, we construct four benchmarks by integrating different types of graph data, e.g., molecules, biological, and social network data. The details are illustrated as follows:

- **MOLECULES:** This benchmark includes multiple molecule datasets, e.g., MUTAG, DHFR, PTC_MR, BZR, COX2, AIDS, and NCI1.
- **BIOCHEM:** This benchmark is a cross-domain dataset including datasets in MOLECULE, and additional biological datasets, e.g., ENZYMES, PROTEINS, and DD.
- **SOCIALNET:** This benchmark includes multiple social network datasets, e.g., IMDB-BINARY, COLLAB and IMDB-MULTI.
- **MIX:** This benchmark contains all datasets from three domains, i.e., molecular, biological, and social network, in Table 4.

Note that we regard the graph in the first class of each dataset as the normal graph, and the graphs in other classes as anomalous graphs. All graph benchmarks used in this paper source from TUDataset [26], a publicly available graph benchmark database[1].

A.2 Detailed Experimental Settings

In this section, we supplement more details of the experimental settings in the paper, including the network structure, trade-off parameter settings, training details, baseline settings, etc.

- **Network Structure:** We employ a 3-layer GIN [37] as the backbone network for our method, with the aggregated dimension in each layer set to 64. In addition, we adopt the 4-layer and 3-layer fully connected networks for the teacher head and student head, respectively. The network structure of the teacher head is set to 256-192-128-64-2, while for the student head is 192-128-64-2. Moreover, we will open-source the code of FGAD for details and reproducibility.
- **Data Split:** For all datasets, we regard the graphs in the first class as normal and graphs in other classes as anomalous. We allocate 80% of the normal graphs data for training, and subsequently construct the testing data by combining the remaining normal data with an equal number of anomalous graphs.
- **Training Details:** We fix the batch size as 64 for all experiments and use Adam [11] as the optimizer with a fixed learning rate $\alpha = 0.001$. We first pre-train each local model excluding the student network and knowledge distillation module for 10 epochs. Then

---

[1]https://chrsmrrs.github.io/datasets/docs/datasets/

Table 4: Detailed information of the datasets used in the experiment.

Dataset Name	#Graphs	#Average Nodes	#Average Edges	#Graph Classes	Data Type
IMDB-BINARY	1,000	19.77	96.53	2	Social Network
COLLAB	5,000	74.49	2,457.78	3	Social Network
IMDB-MULTI	1,500	13.00	65.94	3	Social Network
MUTAG	188	17.93	19.79	2	Molecule
DHFR	756	42.43	44.54	2	Molecule
PTC_MR	344	14.29	14.69	2	Molecule
BZR	405	35.75	38.36	2	Molecule
COX2	467	41.22	43.45	2	Molecule
AIDS	2,000	15.69	16.20	2	Molecule
NCI1	4,110	29.87	32.30	2	Molecule
ENZYMES	600	32.63	62.14	6	Biology
PROTEINS	1,113	39.06	72.82	2	Biology
DD	1,178	284.32	715.66	2	Biology

we jointly train the whole network with collaborative learning for 200 epochs.

- **Trade-off Parameter Settings:** The objective function of FGAD contains two trade-off parameters, i.e., λ, and γ, we vary their values within the range of $[1e^{-4}, 1e^3]$ and evaluate their impact on performance in the Section 4.5.1. Regarding the number of clients C in a single-dataset, we vary it within the range of $[2, \ldots, 10]$ and evaluate its impact in Section 4.5.2, while for multi-dataset, the number of clients is set to the number of its sub-datasets. Besides, for the number of GIN layers K, we also evaluate its impact under different values in Section 4.5.3.
- **Baseline Settings:** For the state-of-the-art baselines including FedAvg, FedProx, GCFL, and FedStar, we integrate them with DeepSVDD [29] to construct the end-to-end GAD model. We also include the self-training strategy that abandons collaborative learning, as one of the baselines. Besides, we employ the same GIN backbone with FGAD to guarantee the fairness of the experiment. The objective of local models in each client is to minimize the distance from the projection of the training data in the latent space to the centroid, which is randomly initialized following the setting in DeepSVDD and fixed throughout the training phase. In the collaborative learning phase, we upload the learned decision boundaries in each client as part of the parameters and aggregate them in the server. Finally, we can calculate the anomaly score by the distances between the graph representation and the centroid after training, and the smaller the score, the more the graph tends to be considered normal.
- **Implementation:** The implementation of FGAD is based on PyTorch Geometric [7] library, and the experiments are run on NVIDIA Tesla A100 GPU with AMD EPYC 7532 CPU.

A.3 Theoretical Complexity Analysis

Here we provide theoretical complexity analysis of the proposed FGAD method. Assume there are N graphs across all clients, and with maximal m nodes and $|E|_{\max}$ edges within a graph. In the local model of each client, the maximal dimension among input and latent space of GIN is denoted by $\tilde{d}$, and the number of GIN layers is

represented by L. In Addition, the maximal latent dimensions of the teacher and student heads are denoted by d_t and d_s, respectively. Besides, the number of latent layers in the teacher and student heads is denoted by K_t and K_s. Subsequently, we analyze the time and space complexity of FGAD within a single client, as well as the communication complexity in collaborative learning, as follows:

- **Time Complexity**: Since the teacher and student models share the same GIN backbone, the time complexity of the backbone network is $O(NL(m\tilde{d}^2 + |E|_{\max}\tilde{d}))$. Similarly, the time complexity of the anomaly generator in the teacher model mainly comes from the GIN. For the teacher and student heads, the time complexities are $O(K_t\tilde{d}d_t)$ and $O(K_s\tilde{d}d_s)$, respectively. Consequently, the overall time complexity of FGAD framework is approximately $O(2NL(m\tilde{d}^2 + |E|_{\max}\tilde{d}) + (K_td_t + K_sd_s)\tilde{d})$, where includes the anomaly generator weight-shared GIN backbone, and the teacher and student heads.
- **Space Complexity**: For the space complexity of the GIN backbone, the space complexity mainly comes from the storage of weight and bias matrices in each layer, which can be denoted by $O(L\tilde{d}(1 + \tilde{d}))$. For the teacher and student heads, their space complexities can be derived similarly, i.e., $O(K_t\tilde{d}(1+d_t)+K_s\tilde{d}(1+d_s))$. Consequently, the overall space complexity of FGAD framework is approximately $O(L\tilde{d}(1 + \tilde{d}) + K_t\tilde{d}(1 + d_t) + K_s\tilde{d}(1 + d_s))$.
- **Communication Complexity**: Since the teacher model in FGAD is used for the personalization of local clients, only the student head engages in collaboration. Consequently, the time and space complexities in a communication round are approximately $O(K_s\tilde{d}d_s)$ and $O(K_s\tilde{d}(1 + d_s))$.

A.4 Empirical Complexity Analysis

To more comprehensively analyze the complexity of FGAD, we further provide empirical complexity analysis. Specifically, we compare the running time (in local) and communication time (in collaboration) of FGAD with other baselines. Note that the experiment is conducted under uniform device settings (detailed in Appendix
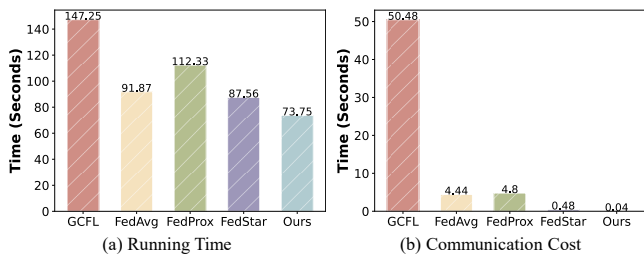
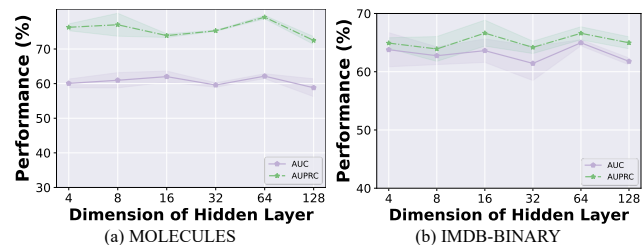**Figure 7: Running time and communication cost comparison in 200 epochs.**

Figure 8: Parameter sensitivity of different dimensions for hidden layers.

A.2) to ensure fairness. The experimental results are presented in Fig. 7.

It can be observed that the time complexity of FGAD is competitive with several baselines, e.g., that of FedStar, and significantly better than that of GCFL. Combined with the performance comparison in Tables 1 and 2 (in the paper), the overall experimental results demonstrate that FGAD not only significantly improves anomaly detection performance but also possesses promising time efficiency compared to other baselines.

Additionally, communication cost (time) is also an important evaluation metric in federated learning. Therefore, we further conduct the comparative experiment to demonstrate the effectiveness of FGAD. As shown in Fig. 7 (b), FGAD has the lowest communication time compared with other baselines, which aligns with the comparison of exchanging amount of network parameters in Table 1. It should be noted that this is the analog communication time without considering the network bandwidth. When it comes to real-world collaboration, the network bandwidth will significantly impact the efficiency of model parameter transmission. Consequently, in cases of models with large parameter sizes, the communication time becomes a pivotal factor influencing the time complexity of collaborative learning.

A.5 Impact of Latent Dimensions

Here, we further conduct additional parameter analysis for the impact of the latent dimension in the GIN layer. Specifically, we set the latent dimension from [4, 128], and the experimental results on MOLECULES and IMDB-BINARY shown in Fig. 8. The results suggest that FGAD exhibits relatively stable performance across a wide range of latent layer dimensions, demonstrating its robustness. Nevertheless, it can be observed that excessively high dimensions

(e.g., 128) might adversely affect performance, potentially due to the redundant information it brings.

A.6 Justification of the Backbone Sharing

To justify the rationale for sharing the backbone network between the teacher and student models, we conduct additional experiments by comparing the performance of FGAD with and without sharing the GIN backbone. The results are presented in Table 5. We can observe only a marginal difference in performance between these two strategies. This observation suggests that sharing the GIN backbone would not decrease the effectiveness of knowledge distillation in FGAD. More importantly, the significant benefit of sharing the GIN backbone is the substantial reduction in model complexity. This streamlined architecture leads to a more efficient model in terms of computational resources and memory usage.

Table 5: Performance (mean(%) ± std(%)) of FGAD under shared/unshared GIN backbone.

Backbone	IMDB-BINARY		IMDB-MULTI	
	AUC	AUPRC	AUC	AUPRC
Shared GIN	64.97±0.52	66.60±1.12	60.51±1.18	66.82±0.14
w/o Shared GIN	63.13±1.19	66.43±2.23	58.13±0.84	66.67±0.00