Extracting Privacy-Preserving Subgraphs in Federated Graph Learning using Information Bottleneck

Chenhan Zhang
chenhan.zhang@student.uts.edu.au
University of Technology Sydney
Sydney, Australia

Weiqi Wang
Weiqi.Wang-2@student.uts.edu.au
University of Technology Sydney
Sydney, Australia

James J.Q. Yu
yujq3@sustech.edu.cn
Southern University of Science and Technology
Shenzhen, China

Shui Yu
shui.yu@uts.edu.au
University of Technology Sydney
Sydney, Australia

ABSTRACT

As graphs are getting larger and larger, federated graph learning (FGL) is increasingly adopted, which can train graph neural networks (GNNs) on distributed graph data. However, the privacy of graph data in FGL systems is an inevitable concern due to multiparty participation. Recent studies indicated that the gradient leakage of trained GNN can be used to infer private graph data information utilizing model inversion attacks (MIA). Moreover, the central server can legitimately access the local GNN gradients, which makes MIA difficult to counter if the attacker is at the central server. In this paper, we first identify a realistic crowdsourcing-based FGL scenario where MIA from the central server towards clients' subgraph structures is a nonnegligible threat. Then, we propose a defense scheme, *Subgraph-Out-of-Subgraph* (SOS), to mitigate such MIA and meanwhile, maintain the prediction accuracy. We leverage the information bottleneck (IB) principle to extract task-relevant subgraphs out of the clients' original subgraphs. The extracted IB-subgraphs are used for local GNN training and the local model updates will have less information about the original subgraphs, which renders the MIA harder to infer the original subgraph structure. Particularly, we devise a novel neural network-powered approach to overcome the intractability of graph data's mutual information estimation in IB optimization. Additionally, we design a subgraph generation algorithm for finally yielding reasonable IB-subgraphs from the optimization results. Extensive experiments demonstrate the efficacy of the proposed scheme, the FGL system trained on IB-subgraphs is more robust against MIA attacks with minuscule accuracy loss.

CCS CONCEPTS

• **Security and privacy**; • **Computing methodologies** → *Machine learning*;

KEYWORDS

Graph Neural Networks; Federated Learning; Model Inversion Attack; Information Bottleneck

ACM Reference Format:
Chenhan Zhang, Weiqi Wang, James J.Q. Yu, and Shui Yu. 2022. Extracting Privacy-Preserving Subgraphs in Federated Graph Learning using Information Bottleneck. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (AsiaCCS '23).* ACM, New York, NY, USA, 13 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Federated learning (FL) has been recognized as a privacy-preserving solution to the data-silo problem in machine learning [43]. With graph neural networks (GNN) hitting the mainstream of ML, FL is extended to the graph domain, i.e., federated graph learning (FGL). Privacy preservation in FGL is more challenging than that in conventional FL. In addition to feature attributes, the attributes of graph structure are also considered as data contributors' sensitive information. A graph structure can store the personal social relationships or commercial intercourse among different nodal entities (e.g., transaction network), and it can also imply the intellectual property of a data contributor, as (s)he may consume massive resources on collecting the relationships among different nodal entities (e.g., citation network) [9, 14].

In crowdsourcing-based FGL systems, the graph structures are intellectual properties of workers (clients), as they can be the products of workers in fine-processing the assigned coarse graph data (refer to Figure 2 in Section 3.2). Therefore, the privacy of graph structures should be emphasized. However, vanilla FGL systems can only guarantee clients' subgraph data privacy when the FL protocol is strictly followed. Recent studies indicated that model inversion attacks (MIA) can be utilized to steal private graph structures from the leaked gradients or outputs of trained GNN models [9, 54], which can be regarded as a big threat to the privacy of graph data. It is even more difficult to identify MIA against FGL systems: as the central server can legitimately access the local GNN updates according to the FL protocol, it can further infer the corresponding private data information from any local GNN update, intentionally and stealthily (see Figure 1).

As FGL systems are usually adopted in scenarios in which the privacy of graph data is a major concern, the privacy-preserving capability of FGL systems should be emphasized, however, not yet

AsiaCCS '23, July 10–14, 2023, Melbourne, Australia
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

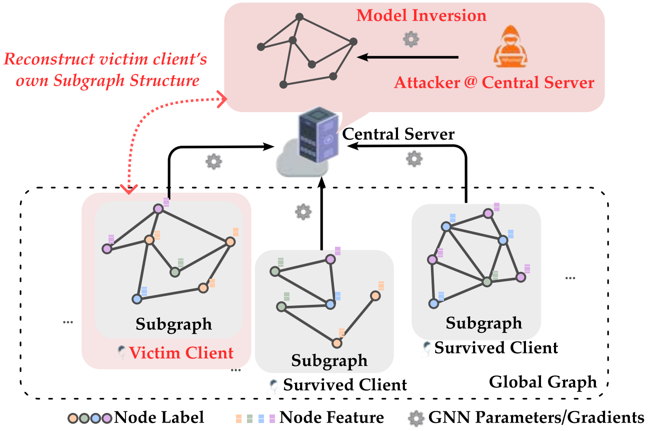

Figure 1: Illustration of a model inversion attack launched by an attacker at the central server in a subgraph-level FGL system, which attempts to reconstruct the graph structure of a client's subgraph.

in existing FGL approaches [4]. Among the works addressing the privacy issues of GNN and FGL systems, differential privacy (DP) is primarily regarded as the countermeasure, which adds calibrated noise to nodal attributes [40, 51]. However, research has shown that neural network-based reconstructing attacks can circumvent DP protection to a significant degree [10]. In addition, privacy protection of nodal attributes differs from that of the graph structure we are concerned about. In addition, the use of DP techniques can have many limitations in achieving a satisfactory data utility-privacy tradeoff [21, 53].

From another perspective, the existing approaches to FGL are usually designed in an *incremental manner* to improve the learning capacity. For example, in [49], the missing edges connecting different subgraphs are considered to be significant for improving the connection among different local GNN updates. Therefore, each client is required to additionally train a generator to recover missing neighbors serving its local GNN training. On the one hand, these processes inevitably increase the communicational and computational burden on the overall system. On the other hand, the learned GNN models in these approaches would record more information related to the original subgraph, naturally expanding the attack surface to MIA. The danger of private subgraph data leakage thus increases. Therefore, we also wonder: *in terms of designing FGL defense schemes, whether it is possible to do "subtraction" rather than do "addition" ?*

One solution using the notion of "subtraction" is data compression. Conventional data compression approaches can be accompanied by a loss of data utility. Compressive privacy (CP) enables the compression to be customized in accordance with known utility and privacy models [26]. Previous research found that the information bottleneck (IB) [31] principle can provide a critical solution for CP: an optimal tradeoff between utility gain and privacy loss [18]. In the domain of ML on regular data, the IB principle has demonstrated its efficacy in privacy preservation since it can work as a "privacy funnel" to distort the information in a new data representation while maintaining its informative (predictive) information [32]. Unlike regular data such as images, graph data are sampled from non-Euclidean space, which renders their unique characteristics such as discreteness due to the existence of edges [49]. Unfortunately, the discreteness makes the direct compression of the graph structure unachievable. Recent studies indicated that the explainability of GNNs is highly associated with some implicit subgraphs of a graph, as they can reveal which components (e.g., motifs) of a graph support the final predictions of GNNs and also which components are related to the privacy of the entire graph structure [4].

In this paper, we proposed to leverage the information bottleneck (IB) principle to identify a smaller subgraph, namely, IB-subgraph, as the "compression" from the client's original subgraph to serve the training of FGL — we name this scheme as *Subgraph-Out-of-Subgraph* (SOS). On the one hand, the obtained IB-subgraphs are as informative as possible concerning the task target (e.g., the node classification labels) in order to develop accurate predictions. On the other hand, they are distorted from the original subgraphs, where the task-irrelevant information can be juiced out as much as possible. In this way, less information related to the original subgraphs' structures would be learned and recorded by the GNN model, which further mitigates MIA on the FGL system. Moreover, the involved IB-subgraph publishing phase is only executed once in a FGL life span, making our scheme computationally inexpensive.

We summarize our main contributions as follows:

- We identify a realistic scenario of MIA in FGL systems where the graph structure owned by the client is under the threat of MIA from the central server.
- To defend FGL against MIA, we propose a novel scheme leveraging the information bottleneck (IB) principle that identifies smaller IB-subgraphs from clients' original subgraphs for local GNN training.
- We further devise a neural network-powered approach to estimate mutual information for IB optimization. We also design an algorithm to reasonably construct the final IB-subgraphs from the optimization results. The developed IB-subgraphs are distorted yet informative.
- We conduct comprehensive case studies on three graph datasets. The results show that the IB-subgraphs developed by the proposed approach can better resist MIA yet reach the same prediction accuracy level as the original subgraphs. Moreover, the proposed scheme is computationally efficient compared with other FGL schemes.

The remainder of this paper is organized as follows. In Section 2, we introduce the related work. Section 3 defines the problems of subgraph-level FGL and the threat model to be investigated in this work. We elaborate on the proposed SOS scheme in Section 4 and perform a series of experiments to demonstrate the effectiveness of the proposed approach in Section 5. Section 6 concludes the paper with a discussion of future work.

2 RELATED WORK

2.1 Security and Privacy of Graph Neural Networks

Security and privacy concerns about machine learning and related assets are increasing. A plethora of research has demonstrated that machine learning models are vulnerable to security and privacy attacks, such as adversarial attacks [6], model inversion attacks [5]. As an extension of machine learning to graph-structured data, GNNs are also vulnerable to these attacks [29].

The majority of existing works focus on the aspect of GNN security such as the adversarial attack on GNNs. For example, Zügner *et al.* [57] have shown that both features and structure modification on graph data could significantly reduce GNNs' accuracy. Zhang *et al.* [52] indicated subgraphs can be cast for backdoor attacks on GNNs. With the concern for data privacy in recent years, the privacy of graph data has also started to receive research attention. Particularly, model inversion attacks (MIA) are a nonnegligible threat to the privacy of graph data, which can steal graph data from a trained GNN. Considering a black-box setting, He *et al.* [9] proposed to utilize the outputs of GNN models to infer the graph data the GNN models were trained on. Considering a white-box setting, Zhang *et al.* [54] demonstrated that gradient information of a trained GNN can be utilized to reconstruct the graph with graph autoencoder. Wu *et al.* [38] focused on graph-level MIA, with the aim of identifying if a graph sample was used to train a GNN model. Reference [51] considered the information leakage of the graph embeddings and designed a series of corresponding MIA approaches. Along with MIA attacks, model extraction attacks (MEA) are another typical privacy attack on graph neural networks (GNNs), which focus on the GNN model itself rather than the data, aiming to extract a model that achieves similar performance to the target model [39].

To counter/mitigate these attacks, a few defense methods have been proposed [29]. Most existing studies considered defending against adversarial attacks [41], but less attention has been paid to the prevent the privacy leakage of GNNs. Among the few works concerning the privacy problems of GNN, differential privacy (DP) is mainly considered as the countermeasure, which introduces noise to nodal attributes [40, 51]. However, research has demonstrated that neural network-based reconstructing attacks can bypass the DP protection to a great extent [10]. Furthermore, the privacy preservation of nodal attributes is still far from the one of the graph structure.

Therefore, we are motivated to study the countermeasures against model inversion attacks — one of the most threatening privacy attacks. Furthermore, we consider white-box attacks in this work: the attackers know more information about the target than it is in black-box settings, which could be more aggressive.

2.2 Information Bottleneck of Graph Data

Information bottleneck (IB) was first proposed for preserving the maximum *mutual information* (MI) of input data during encoding [31]. Alemi *et al.* [1] first extended IB to deep learning by the proposed variational information bottleneck (VIB). The capacity to extract condensed and significant representation makes it a promising tool for enhancing learning performance in various learning tasks [36]. Nevertheless, conventional IB methods like VIB cannot handle graph data due to the intractability of MI calculation for irregular data. To address the problems, some earlier studies such as [34], [24], and [28] adopted MI maximization [11] to obtain graph representations. Wu *et al.* [42] first conceptualized the *graph representation learning with IB principle* as graph information bottleneck (GIB), and leveraged Gaussian prior assumption to sample neighbors for node representation learning. Our notion of IB-based subgraph extraction is akin to the subgraph recognition pattern in [45], however, the one in [45] is designed for graph classification tasks. Furthermore, for most of GIB studies [24, 28, 34, 45], Security and privacy robustness of the IB-extracted graph representations is not involved. While [42] showed that the graph representation attained by the proposed GIB could be resilient to adversarial attack, the robustness to model inversion attack is not investigated in this work.

As an extension and supplement to the body knowledge, this work is dedicated to adapting the information bottleneck to solve the privacy problem of graph data in a realistic scenario. This methodology can be associated with the explainability of privacy issues in graph deep learning [47].

2.3 Federated Graph Learning

Federated learning (FL) is an emerging machine learning technique that allows decentralized and privacy-preserving learning among multiple parties [43]. While FL has been extensively investigated with Euclidean data (e.g., images), the cases regarding irregular graph data are still in the early stage. Generally, there are three types of FGL scenarios, namely, *Node-level* FL [15], *Graph-level* FGL [8], and *Subgraph-level* FGL [49]. Particularly, our work falls into the realm of subgraph-level FGL in which each client holds a subgraph that is part of a larger global graph.

The challenge of non-independent and identically distributed (non-IID) datasets in conventional FL training becomes more severe in the context of FGL. In addition to features and labels, heterogeneous graph structure (topology) distributions could also have a great impact on FL training. Specific to subgraph-level FGL, Zheng *et al.* [56] exploited the distribution divergence among different clients' datasets using split learning. Zhang *et al.* [49] mitigated the issue by leveraging GraphSAGE [7] model to improve the inductiveness and scalability of graph mining. Moreover, the authors attached importance to the missing connections among subgraphs and tried to recover them as they may contain important information bridging different clients.

However, the effectiveness of these approaches is based upon additional and iterative operations executed by the central server and clients, which introduce not a small communicational and computational burden [35]. Credited to the IB principle, our proposed scheme highlights the overall performance of the FGL, including the tradeoff between privacy and accuracy, and computational efficiency.

3 PRELIMINARY

3.1 Subgraph-level Federated Learning for Node Classification Tasks

Our work focuses on the subgraph-level FGL system since it is the best match for the investigated privacy attack scenario. Subgraph-level federated learning assumes that horizontally distributed subgraphs of a global graph are held by clients. Thus, in the subgraph-level FGL system, we have the central server S and K clients with distributed subgraphs. Given a global graph $G = \{\mathcal{V}, \mathcal{E}, X\}$, each

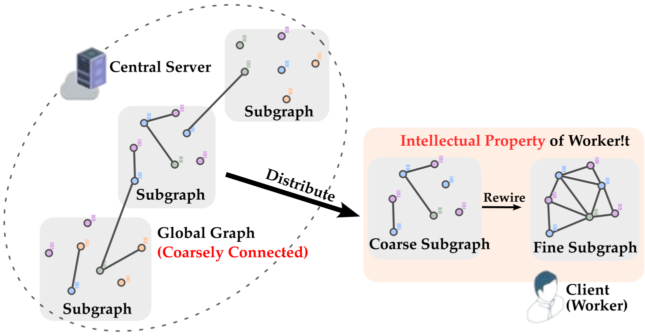

Figure 2: The central server only has a coarsely connected global graph. The workers have to manually fine-process the subgraph and hence owns the intellectual property of such a subgraph structure.

subgraph is denoted by $G_i := \{\mathcal{V}_i, \mathcal{E}_i, \boldsymbol{X}_i \mid i \in [K]\} := \{\boldsymbol{A}_i, \boldsymbol{X}_i\}$ where $\mathcal{V} = \mathcal{V}_1 \cup \cdots \cup \mathcal{V}_K$, $\boldsymbol{X}_i = \{x_j\} \in \mathbb{R}^{N \times d}$ (d is the feature dimension). Note that unless other stated, we use subscript i as the subgraph-level (client-level) index to distinguish a subgraph (client), and subscript j as the node-level index to distinguish a node.

The semi-supervised node classification task is investigated in this work. Given the set of labeled nodes $\mathcal{V}_l$, the set of unlabeled nodes $\mathcal{V}_u = \mathcal{V} \setminus \mathcal{V}_l$, and the labels $Y = \{y_j\} \in \mathbb{R}^{|\mathcal{V}_l|}$, the task require the FGL system to develop a GNN-based classifier $f : \boldsymbol{X} \rightarrow Y$ that can map the class of each unlabeled node to the exact one. We use $\boldsymbol{W}$ to denote all the learnable parameters (weights) of f.

Further, the goal of such a FGL system is to develop a global GNN model F with optimized parameters $\boldsymbol{W}_s$ that minimizes the prediction loss after aggregation, which can be formulated as

$$\min_{\boldsymbol{W}_s} \mathcal{R}_s\left(F(\boldsymbol{W}_s; \boldsymbol{G})\right) := \min_{\boldsymbol{W}_s} \frac{1}{K} \sum_{i=1}^{K} \mathcal{R}_i\left(f_i(\boldsymbol{W}_s; \boldsymbol{G}_i)\right), \tag{1}$$

where $\mathcal{R}_s$ and $\mathcal{R}_i$ are the global and i-th client's empirical risks, respectively. In this work, FedAvg algorithm [20] is adopted as most of the studies in this domain — S aggregates and averages the clients' local model parameters to obtain the global model. It is also worth mentioning that GNN model is treated inductively in this work — the parameter aggregation process does not require graph structure information.

3.2 Threat Model

In this work, the threat model is assumed in a realistic crowdsourcing-based FGL scenario. The curator of the central server is the initial graph data owner, which splits the global graph data into several subgraph data and distributes them to each client (worker). Therefore, (S)he knows the nodal feature and labels of the distributed data. However, the original global graph only has a *coarse* connection pattern. The workers have to further *rewire* the received subgraphs by collecting/identifying relationships among nodal entities to construct reasonable training sets for the local GNN models — *fine subgraphs*. An illustration is given in Figure 2. Since this process is laborious and resource-consuming for the workers, the graph structures of the subgraphs are naturally part of their intellectual properties. It is worth stating additionally that while some studies suggested that it can be more efficient for workers to learn new graph structures from coarse subgraphs [30, 50], the utility of graph data can be compromised. Therefore, we consider the manual collection of graph construction, as a traditional paradigm, to be of existential significance.

We assume that the attacker is the curator of the central server, which is curious about the subgraph structure $\boldsymbol{A}_i$ constructed by the workers. Since the curator can only access the model updates according to the FL protocol, (s)he attempts to leverage model inversion attacks (MIA) to reconstruct the subgraph structures based on the received model updates and the subgraph attributes (s)he has already known. We know that the model update $\boldsymbol{W}$ is associated with the graph structure $\boldsymbol{A}$, where the former is developed by the latter in local GNN training. Due to this correlation, releasing $\boldsymbol{W}$ to the central server will enable him or her to draw some inferences on $\boldsymbol{A}$. Furthermore, we know that the MIA attack is in a white-box setting in our investigated case. The "malicious central server" assumption is akin to the one in [37]; however, as the attacker in our setting has more knowledge related to the target, the attacks will be much more threatening.

More specifically, such a reconstruction is subsumed to "link stealing attack"[9, 54]: the goal of the attacker is to reconstruct target client b's graph structure by identifying whether there exists an edge between each node pair of the subgraph given its local model updates (parameters) $\boldsymbol{W}_b$, feature information $\boldsymbol{X}_b$, and label information Y_b. The attack process can be formulated by

$$\max_{\boldsymbol{A}} P\left(\boldsymbol{A} \mid \boldsymbol{X}_b, \boldsymbol{Y}_b, \boldsymbol{W}_b\right), \tag{2}$$

where P is the posterior possibility that the attacker aims to maximize by finding the adjacency matrix $\boldsymbol{A}$.

To mitigate the inference threat on $\boldsymbol{A}$, the client can release a distorted version of either 1) $\boldsymbol{A}$ before local GNN training or 2) $\boldsymbol{W}$ before model updating. In this paper, we adopt the first notion to construct our approach. Additionally, we also compare with a local differential privacy (LDP) method who adopts the second notion in our experiments.

3.3 Information Bottleneck Principle

Let $I(X, Z)$ denote the mutual information (MI) between the input X and the encoded representation Z, and $I(Y, Z)$ denote the MI between Z and the class label Y. IB principle [31] optimizes a tradeoff between $I(Y, Z)$ and $I(X, Z)$, that is

$$\min_{Z} \mathcal{L}_{\mathrm{IB}} = -I(Y, Z) + \beta I(X, Z), \tag{3}$$

where X is the input and Z is the encoded representation of X, Y is the class label, and β is a Lagrange multiplier to control the compression extent of Z.

Alemi [1] *et al.* proposed a variational approximation to Eq (3) by using a neural network to parameterize the distribution, known as variational information bottleneck (VIB), which is defined as

$$\begin{aligned} \mathcal{L}_{\mathrm{VIB}} = \frac{1}{N} \sum_{i=1}^{N} \int & p(z \mid x_i) \log q_\phi(y_i \mid z)\, dz \\ & -\beta \mathrm{KL}\left(p(z \mid x_i) \mid r(z)\right), \end{aligned} \tag{4}$$

where $q_\phi(y_i \mid z)$ is the variational approximation to $p(y_i \mid z)$, $r(z)$ is the variational approximation of $p(z)$, and $\mathrm{KL}(\cdot)$ denotes the Kullback–Leibler divergence (KLD).

4 SOS: PROPOSED SCHEME

In this section, we present the proposed SOS scheme in a top-down manner. We commence from a framework view by introducing the IB-subgraph publishing mechanism for subgraph-level FGL systems. Then, we define the graph information bottleneck in subgraph-level FGL and elaborate on the Sub-GIB approach from mutual information estimation to the final IB-subgraph generation. Lastly, we discuss the privacy and utility of IB-subgraphs.

Algorithm 1: Brief pipeline of SOS

```
Input: Number of clusters $K$, subgraphs
       $G_i = \{\mathcal{V}_i, \mathcal{E}_i, X_i \mid i \in [K]\}$, GNN model $f$, learning
       rate $\eta$, number of IB optimization epochs $T_{\mathrm{IB}}$,
       number of global training epochs $T$, number of local
       training epochs $T_l$.
Output: $\mathcal{R}_s(F(W_s; G))$.
// Phase 1: IB-subgraphs Publishing
foreach $i \in K$ in parallel do
    foreach epoch $t = 1, 2, \ldots, T_{IB}$ do
        $\theta_{\mathrm{h}}, \theta_{\mathrm{r}}, \theta_{\mathrm{g}}, \theta_{\mathrm{b}} \leftarrow$ Optimize $\mathrm{NN_h}, \mathrm{NN_r}, \mathrm{NN_l}, \mathrm{NN_b}$ via
          Eq. (13)
    $B_i \leftarrow \mathrm{NN_b}(\theta_{\mathrm{b}}; \mathrm{GNN}(G_i))$
    $G_i^{\mathrm{IB}} \leftarrow$ Construct IB-subgraph via the algorithm
      described in Section 4.3 with $B_i$
// Phase 2: Federated Graph Learning on IB-subgraphs
$W_{i,0} \leftarrow$ Initialize GNN model $f$
foreach epoch $t = 1, 2, \ldots, T$ do
    # Updates local GNN model's weights
    foreach $i \in K$ in parallel do
        foreach epoch $e = 1, 2, \ldots, T_l$ do
            $W_{i,t} \leftarrow W_{i,t-1} - \eta \cdot \nabla \mathcal{L}(f(G_i^{\mathrm{IB}}), Y_i)$
    # Updates aggregation at central server (FedAvg) and
      broadcasting
    $W_{s,t} \leftarrow \frac{1}{K} \sum_{i=1}^{K} W_{i,t}$
return $\mathcal{R}_s(F(W_s; G))$
```

4.1 IB-subgraph Publishing Mechanism for Subgraph-level Federated Learning Systems

As shown in Figure 3 and Algorithm 1, the proposed SOS scheme introduces two phases for subgraph-level FGL systems: (1) IB-subgraph publishing; (2) FGL on IB-sbugraphs. In reality, the global graph data can be either static or dynamic. We consider the graphs in a static condition as a single FGL lifespan. That is to say: if any of the subgraphs changes, the previous lifespan is over, and it will proceed to the subsequent FGL lifespan. The proposed approach is incorporated as an *in-processing* one catered for a *single* FGL lifespan.

In the IB-subgraph publishing phase, the system requires each client to generate an IB-subgraph out of the original subgraph. Specifically, we propose the subgraph generation with information bottleneck (Sub-GIB) approach for IB-subgraph generation, which will be introduced later. Given an attributed graph $G = (A, X)$, we name its subgraph developed by Sub-GIB as IB-subgraph denoted by G^{IB}.

Once all the IB-subgraphs are published, the system will proceed to the FGL phase. The clients will hold their original subgraphs and IB-subgraphs locally. Notably, for privacy concerns, the clients will train their local GNN model on the IB-subgraphs instead of the original subgraph. Thus, original subgraphs will not be involved in the FL training. This design embodies the advantage of the proposed approach: as the local GNN model is trained on IB-subgraph data, less "footprint" associated with the original subgraph will be left on the local model updates. As thus, if the central server or any outsider (if there is a gradient leakage) intends to infer the original subgraph information (cf. the model inversion attack scenario described in Section 3.2), the inference effect would be impeded to a great extent. Another advantage is that according to the FL protocol, ones cannot know whether local models were trained on the original subgraphs or any processed subgraphs. Even if they successfully reconstruct the training graphs (i.e., IB-subgraphs), they are not identical to the original subgraphs.

Furthermore, the IB-subgraph publishing phase is designed to be *one-off* in a FGL lifespan — it will only be executed in the initializing stage for one time. As the subgraphs are all static in a lifespan, it is of no necessity to publish their IB-subgraphs iteratively along with the training epochs. One may notice that the FGL phase in the proposed scheme remains the same as the naive FedAvg algorithm, it does not introduce any additional actions to both the central server and clients. These designs endow the proposed scheme with some advantages. First, the algorithmic complexity of the proposed scheme is close to that of vanilla FGL, making it computationally and communicationally sparing. Second, The proposed scheme is both model-agnostic and FL algorithm-agnostic, which can be integrated with different FL algorithms. Moreover, the proposed scheme is orthogonal to many of the existing subgraph-level FGL approaches, such as the splitting learning-based approach in [56], which can be orchestrated to improve the FGL performance further.

4.2 Sub-GIB: Subgraph Generation with Information Bottleneck

4.2.1 Subgraph Information Bottelneck in Graph Data. Generally, Sub-GIB extends the IB principle, which casts about for most predictive but compressed G^{IB} by: (1) minimizing the MI between G^{IB} and G, i.e., $I\left(G, G^{\mathrm{IB}}\right)$); (2) maximizing the MI between G^{IB} and Y, i.e., $I\left(Y, G^{\mathrm{IB}}\right)$. The optimization objective of Sub-GIB can be formulated as

$$\min_{G^{\mathrm{IB}} \in \mathbb{G}^{\mathrm{IB}}} \mathcal{L}_{\mathrm{GIB}} = -I\left(Y, G^{\mathrm{IB}}\right) + \beta I\left(G, G^{\mathrm{IB}}\right), \tag{5}$$

where $\mathbb{G}^{\mathrm{IB}}$ denotes the subgraph search space of G.

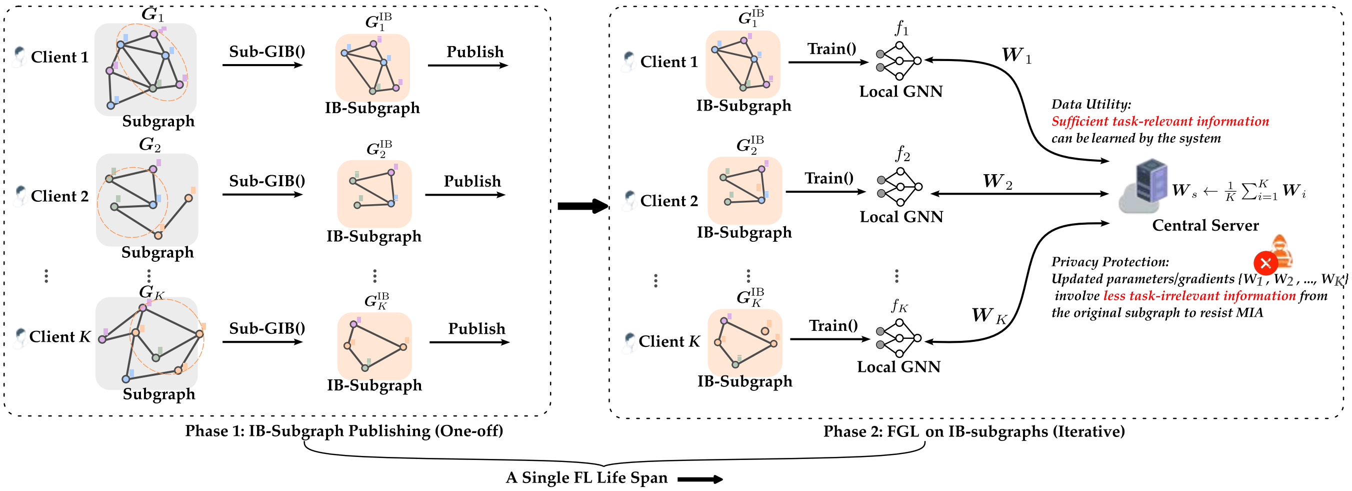

Figure 3: Schematic of the proposed SOS scheme.

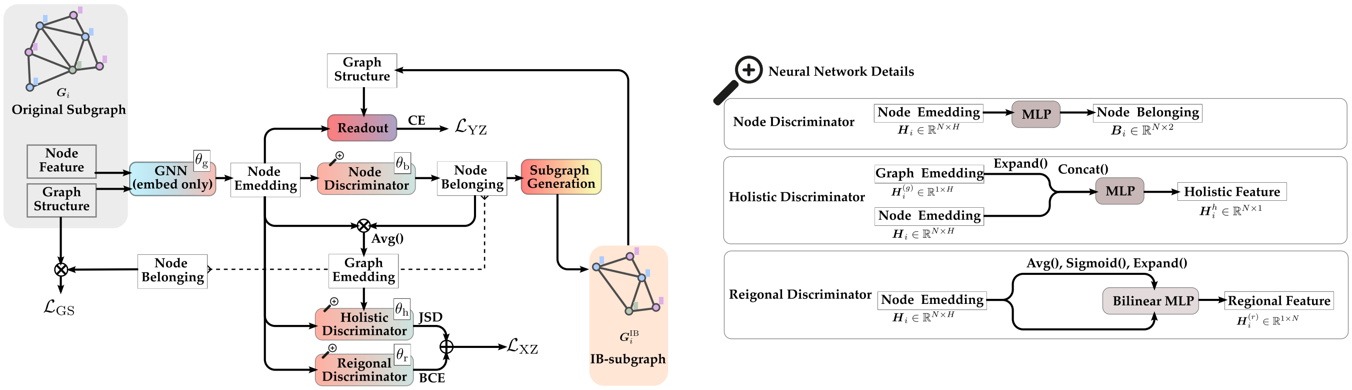

Figure 4: Architecture of Sub-GIB. GNN$(\cdot)$ **and** Readout$(\cdot)$ **operations differ from the practically adopted GNN models in the FL systems. Negative samples for the holistic discriminator and regional discriminator are generated by row-wise shuffling the feature matrix X_i but keep the original adjacency matrix, i.e., $\tilde{G}_i = (A_i, \tilde{X}_i)$.**

We generalize the Sub-GIB for the learning of a centralized GNN to the subgraph-level FGL scenario. In contrast to centralized systems, the FGL system involves multiple clients, each of which already owns a smaller subgraph of a global graph. In such a federated scenario, Sub-GIB seeks to further recognize a subgraph of the subgraph owned by these clients — we name this process as *Subgraph-Out-of-Subgraph* (SOS). Each IB-subgraph is expected to embody *minimal-sufficient* information. *Sufficient* requires that IB-subgraph is as informative as possible regarding the target to develop accurate predictions. *Minimal* promotes the IB-subgraph to be distorted from the original subgraph, and the information that is irrelevant to the prediction can be juiced out as much as possible. In this work, we in particular leverage the *Minimal* feature to achieve the privacy preservation of the IB-subgraph.

Furthermore, we only consider the graph structural compression in this work. That is to say that the encoded G_i^{IB} will have a new and smaller graph structure; however, the nodal feature will not be encoded. As shown in Figure 3, the attributes of the nodes that are eliminated after the compression will be naturally discarded.

4.2.2 Neural Network-powered Mutual Information Estimation.

In this work, we commence by using VIB [1] to optimize the Sub-GIB for each client's IB-subgraph generation. Describing the information transmission among Y, G, and G^{IB}, the Markov chain in VIB is

$$Y \overset{p(G|y)}{\longrightarrow} G \overset{q_{\theta_2}(G^{\mathrm{IB}}|G)}{\longrightarrow} G^{\mathrm{IB}} \overset{q_{\theta_1}(y|G^{\mathrm{IB}})}{\longrightarrow} \hat{Y}, \tag{6}$$

where $q_{\theta_1}\left(y \mid G^{\mathrm{IB}}\right)$ and $q_{\theta_2}(G^{\mathrm{IB}} \mid G)$ are reparameterized variational approximations to $p\left(y \mid G^{\mathrm{IB}}\right)$ and $p\left(G^{\mathrm{IB}} \mid G\right)$, respectively.

Resorting to the deduction in [46], a tractable variational lower bound for the first term of Eq. (5) for each client i can be derived as

$$\begin{aligned} I\left(Y_i, G_i^{\mathrm{IB}}\right) &= \int p\left(y, G_i^{\mathrm{IB}}\right) \log p\left(y \mid G_i^{\mathrm{IB}}\right) dy\, dG_i^{\mathrm{IB}} + H(Y_i) \\ &\geq \int p\left(y, G_i^{\mathrm{IB}}\right) \log q_{\theta_1}\left(y \mid G_i^{\mathrm{IB}}\right) dy\, dG_i^{\mathrm{IB}}, \end{aligned} \tag{7}$$

where the entropy of labels $H(Y_i)$ can be neglected as it is independent of the optimization. In the practical optimization, Eq. (7) can be transformed into the node classification loss between the real labels Y_i and the labels predicted on G_i^{IB}, which can be formulated as

$$\min_{\boldsymbol{G}_i^{\mathrm{IB}}} \mathcal{L}_{\mathrm{YZ}} = \mathcal{L}_{\mathrm{ce}}(f_{\theta_{\mathrm{g}}}(\boldsymbol{G}_i^{\mathrm{IB}}), \boldsymbol{Y}_i), \tag{8}$$

where $f_{\theta_{\mathrm{g}}}(\cdot)$ is the adopted GNN-based classifier in the FGL system, and $\mathcal{L}_{\mathrm{ce}}(\cdot)$ denotes the cross entropy loss. As some of the nodes in $\boldsymbol{G}_i$ are eliminated in $\boldsymbol{G}_i^{\mathrm{IB}}$, the loss evaluation are only performed on the nodes included in $\boldsymbol{G}_i^{\mathrm{IB}}$.

The second term of Eq. (4) is tractable for the primitive VIB only if the data's empirical distribution information is known to further compute the MI. Nevertheless, the discreteness and non-IID of graph-structured data renders its empirical distribution untraceable [42]. In other words, we cannot find a proper prior distribution $q_{\theta_2}(\boldsymbol{G}^{\mathrm{IB}})$ (cf. $r(z)$ in Eq. (4)) for $\boldsymbol{G}^{\mathrm{IB}}$.

To make $I\left(\boldsymbol{G}_i, \boldsymbol{G}_i^{\mathrm{IB}}\right)$ tractable, we introduce a neural network-powered mutual information estimation approach to instantiate and minimize $I\left(\boldsymbol{G}_i, \boldsymbol{G}_i^{\mathrm{IB}}\right)$. We adapt ideas from [11] to consider both *holistic* and *regional* features[1]. We first introduce a neural network-based extractor $f_h(\cdot)$ to learn implicit graph-level representation (holistic feature). Basically, $f_h(\cdot)$ consists of an encoder and a discriminator $f_{\theta_{\mathrm{h}}}(\cdot)$. The encoder here shares the same architecture and parameters with $f_{\theta_{\mathrm{g}}}(\cdot)$ in Eq. (8). Thus, we have $f_h = f_{\theta_{\mathrm{g}}} \circ f_{\theta_{\mathrm{h}}}$. Particularly, we define holistic feature developed by $f_h(\cdot)$ as the graph-level information — summarizing the graph feature as a whole. Different from prior work [30, 46], we propose to use Jensen-Shannon divergence (JSD) to evaluate MI, which requires a smaller number of negative samples and demonstrates better stability in practice. Based on a JSD-based MI estimation[23], we can formulate the objective specific to the holistic feature as

$$\begin{aligned}\mathcal{L}_h = &-\log \mathbb{E}_{\tilde{\boldsymbol{G}}_i \in p(\boldsymbol{G}_i, \boldsymbol{G}_i^{\mathrm{IB}})}\left(1+e^{f_h\left(\tilde{\boldsymbol{G}}_i\right)}\right) \\ &-\log \mathbb{E}_{\boldsymbol{G}_i \in p(\boldsymbol{G}_i)}\left(1+e^{-f_h(\boldsymbol{G}_i)}\right),\end{aligned} \tag{9}$$

where $\tilde{\boldsymbol{G}}_i$ represents the negative samples from the joint distribution $p\left(\boldsymbol{G}_i, \boldsymbol{G}_i^{\mathrm{IB}}\right)$, which is instantiated by row-wise shuffling the feature matrix $\boldsymbol{X}_i$ but keep the original adjacency matrix, i.e., $\tilde{\boldsymbol{G}}_i = (\boldsymbol{A}_i, \tilde{\boldsymbol{X}}_i)$.

We consider regional features as specific node-level representations contributing to the node classification. Correspondingly, let $f_r(\cdot)$ be the neural network-based extractors for the regional feature. Similar to $f_h(\cdot)$, $f_r(\cdot)$ adopts $f_{\theta_{\mathrm{g}}}(\cdot)$ as the encoder but with a specific discriminator $f_{\theta_{\mathrm{r}}}(\cdot)$, i.e., $f_r = f_{\theta_{\mathrm{g}}} \circ f_{\theta_{\mathrm{r}}}$. The details of $f_{\theta_{\mathrm{r}}}(\cdot)$ and $f_{\theta_{\mathrm{r}}}(\cdot)$ are illustrated in Figure 4. Incorporating the negative samples, we use binary cross-entropy (BCE) to evaluate the loss of regional MI, which can be formulated as

$$\mathcal{L}_r = \mathcal{L}_{\mathrm{ce}}(f_r(\boldsymbol{G}_i), \mathbf{1}) + \mathcal{L}_{\mathrm{ce}}(f_r(\tilde{\boldsymbol{G}}_i), \mathbf{0}), \tag{10}$$

where $\mathbf{1}$ and $\mathbf{0}$ are the all-one and all-zero vectors respectively representing the positive and negative labels. Combining Eq. (9) and (10), we can obtain the objective estimating $I\left(\boldsymbol{G}_i, \boldsymbol{G}_i^{\mathrm{IB}}\right)$:

$$\min_{\theta_{\mathrm{h}}, \theta_{\mathrm{r}}} \mathcal{L}_{\mathrm{XZ}} = \mathcal{L}_h + \gamma \mathcal{L}_r \tag{11}$$

where γ is a multiplier to control the tradeoff between the two parts. This design is different from the one in [46] which only considers the holistic feature for the graph classification task. We believe that involving holistic features and regional feature will further improve the quality of generated IB-subgraphs, and the hypothesis is justified in our experiments.

To ensure that $\boldsymbol{G}_i^{\mathrm{IB}}$ has a compact graph structure with sufficient feature smoothness, we additionally introduce a loss term with respect to the generated graph structure based on the one in [54], which is formulated as

$$\min_{\theta_{\mathrm{h}}, \theta_{\mathrm{r}}} \mathcal{L}_{\mathrm{GS}} = \mathrm{Tr}(\boldsymbol{B}_i^T \boldsymbol{L}_i \boldsymbol{B}_i), \tag{12}$$

where $\boldsymbol{L}_i$ is the Laplacian adjacency matrix of $\boldsymbol{G}_i$, $\boldsymbol{B}_i$ is a node belonging of $\boldsymbol{G}_i^{\mathrm{IB}}$ (the computation will be detailed in Section 4.3), and $\mathrm{Tr}(\cdot)$ represents the trace of a matrix.

Combining Eq. (8), (11), and (12), we can obtain the final objective function of Sub-GIB, that is

$$\min_{\theta_{\mathrm{h}}, \theta_{\mathrm{r}}, \theta_{\mathrm{g}}, \theta_{\mathrm{b}}} \mathcal{L}_{\text{Sub-GIB}} = \mathcal{L}_{\mathrm{YZ}}(\theta_{\mathrm{g}}, \theta_{\mathrm{b}}) + \beta \mathcal{L}_{\mathrm{XZ}}(\theta_{\mathrm{h}}, \theta_{\mathrm{r}}) + \mathcal{L}_{\mathrm{GS}}(\theta_{\mathrm{g}}, \theta_{\mathrm{b}}). \tag{13}$$

In practice, Eq. (13) is optimized in a bi-level manner: we first optimize $\mathcal{L}_{\mathrm{XZ}}(\theta_{\mathrm{h}}, \theta_{\mathrm{r}})$ and fixed θ_{h} and θ_{r} to further optimize Eq. (13) as a whole. Corresponding to Eq. 6, we have $\theta_1 = \{\theta_{\mathrm{g}}\}$ and $\theta_2 = \{\theta_{\mathrm{h}}, \theta_{\mathrm{r}}, \theta_{\mathrm{b}}\}$

4.3 IB-subgraph Generation Algorithm

We design an algorithm as the last step of Sub-GIB to generate the $\boldsymbol{G}_i^{\mathrm{IB}}$ for publishment. When Sub-GIB is optimized, we first use a node discriminator (denoted by $\mathrm{NN_b}(\cdot)$) to generate a node belonging to each node, as shown in Figure 4. The node belonging evaluates each node by two score: $\mathrm{IN} \in [0, 1]$ and $\mathrm{OUT} = 1 - \mathrm{IN}$ which is the probability of the node should be included or not included in $\boldsymbol{G}_i^{\mathrm{IB}}$, respectively.

Then, we use Top-K algorithm to sort out all the nodes' IN scores to decide the ones to be reserved in the IB-subgraph. Let ρ be the ratio controlling the number of the nodes that are reserved in $\boldsymbol{G}_i^{\mathrm{IB}}$, we have $N_i^{(\min)} = \mathrm{int}(\rho N_i)$. Thus, the first $N_i^{(\min)}$ points with the largest IN score values in subgraph $\boldsymbol{G}_i$ will be retained. We define a downsampling function $\mathrm{Ds} \in \{-1, +1\}$ to perform this process. The downsampling function is defined as

$$\mathrm{Ds}(j) = \begin{cases} 1 & \text{if } j \in [N_i^{(\min)}] \\ -1 & \text{if } j \notin [N_i^{(\min)}] \end{cases}. \tag{14}$$

We then perform the downsampling operation on the adjacency matrix, which is formulated as

$$\boldsymbol{A}_i^* = \{a_j^*\} = \{\frac{1}{2}(1 + \mathrm{Ds}(j)) a_j\}, \tag{15}$$

where $a_j \in \boldsymbol{A}_i$ and a_j^* are the entry of node j in the adjacency matrix before and after the downsampling, respectively. Then, we can use the updated adjacency matrix $\boldsymbol{A}_i^* \in \mathbb{R}^{N_i^{(\min)} \times N_i^{(\min)}}$ and the corresponding feature matrix $\boldsymbol{X}_i^* \in \mathbb{R}^{N_i^{(\min)} \times d}$ to construct IB-subgraph $\boldsymbol{G}_i^{\mathrm{IB}} = (\boldsymbol{A}_i^*, \boldsymbol{X}_i^*)$.

Such an algorithm ensures that at least $N_i^{(\min)}$ nodes can be retained, eliminating the possibility that no nodes are reserved. The edges connecting the discarded nodes and the retained nodes will be naturally dropped — if some nodes supposed to be retained,

[1] we use "holistic" and "regional" to describe "global" and "local" here to avoid abuse.

Table 1: Statistical summary of Cora, Citeseer, and PubMed datasets.

Data	# Node	# Edge	Density	# Class	# Feature
Cora	2708	5278	$14.3e-4$	7	1433
Citeseer	3312	4536	$8.2e-4$	6	3703
PubMed	19717	44338	$2.2e-4$	3	500

however, become singletons after these edges' dropping, they will be discarded as well.

4.4 Discussion on Privacy and Utility of IB-subgraph

4.4.1 Privacy Analysis. Let $\boldsymbol{W}_i$ and $\boldsymbol{W}_i^{\mathrm{IB}}$ be the local model updates developed by the original subgraph and IB-subgraph, respectively. The privacy leakage can be measured by the mutual information between the local model updates trained on IB-subgraphs and the target private subgraph structure, i.e., $I(\boldsymbol{W}_i^{\mathrm{IB}}, \boldsymbol{A}_i)$. According to the Markov chain stated in Eq. (6), it can be ensured that $\boldsymbol{W}_i^{\mathrm{IB}}$ cannot contain more information about the original subgraph structure $\boldsymbol{A}_i$ than $\boldsymbol{W}_i$ since $\boldsymbol{G}_i = (\boldsymbol{A}_i, \boldsymbol{X}_i)$ subsumes $\boldsymbol{G}_i^{\mathrm{IB}} = (\boldsymbol{A}_i^{\mathrm{IB}}, \boldsymbol{X}_i^{\mathrm{IB}})$. We can further derive the diminishing mutual information with $\boldsymbol{A}_i$ from $\boldsymbol{W}_i$ to $\boldsymbol{W}_i^{\mathrm{IB}}$, i.e.,

$$I\left(\boldsymbol{W}_i, \boldsymbol{A}_i\right) \geq I\left(\boldsymbol{W}_i^{\mathrm{IB}}, \boldsymbol{A}_i\right), \tag{16}$$

where the inequality is irreversible. We can deduce that the upper bound of MIA using $\boldsymbol{W}_i^{\mathrm{IB}}$ is equivalent to MIA using $\boldsymbol{W}_i$. By the same token, the upper bound of MIA using $(\boldsymbol{W}_i^{\mathrm{IB}}, \boldsymbol{X}_i, \boldsymbol{Y}_i)$ is equivalent to MIA using $(\boldsymbol{W}_i, \boldsymbol{X}_i, \boldsymbol{Y}_i)$ As $\boldsymbol{G}_i^{\mathrm{IB}}$ is optimized to maximumly juice out the task-irrelevant MI with $\boldsymbol{G}_i$, MIA using $(\boldsymbol{W}_i^{\mathrm{IB}}, \boldsymbol{X}_i, \boldsymbol{Y}_i)$ will be much less effective.

Moreover, most of the existing privacy-targeted FL approaches modify the model updates to protect the data privacy [2, 55]. While these approaches can help defend against MIA, if the local datasets are actively hacked and data privacy will also be compromised. An inherent advantage of our proposed IB-subgraph is that the publishing mechanism isolates the original subgraphs which are the privacy carriers. For example, the original subgraphs can be further stored in a trusted execution environment [25] at the client side to guarantee privacy protection. This advantage enables the privacy guarantee to hold in broadening threat scenarios.

4.4.2 Utility Analysis. Assume that $\boldsymbol{G}_i^{\mathrm{irr}}$ is the subgraph of $\boldsymbol{G}_i$ which is irrelevant to the target $\boldsymbol{Y}_i$. Following [46], an upper bound for MI between $\boldsymbol{G}_i^{\mathrm{IB}}$ and $\boldsymbol{G}_i^{\mathrm{irr}}$ is derived as

$$I\left(\boldsymbol{G}_i^{\mathrm{irr}}, \boldsymbol{G}_i^{\mathrm{IB}}\right) \leq I\left(\boldsymbol{G}_i, \boldsymbol{G}_i^{\mathrm{IB}}\right) - I\left(\boldsymbol{Y}_i, \boldsymbol{G}_i^{\mathrm{IB}}\right). \tag{17}$$

Eq. (17) proved that $\boldsymbol{G}_i^{\mathrm{IB}}$ is dependent on $\boldsymbol{G}_i^{\mathrm{irr}}$. Thus optimizing Eq. (5) will be equivalent to minimize $I\left(\boldsymbol{G}_i^{\mathrm{irr}}, \boldsymbol{G}_i^{\mathrm{IB}}\right)$, making the optimized $\boldsymbol{G}_i^{\mathrm{IB}}$ would be with less irrelevant information to target $\boldsymbol{Y}_i$.

Table 2: Model Inversion Attack against Different Clients.

		Cora		Citeseer		PubMed	
		AUC	AP	AUC	AP	AUC	AP
G_1	SOS-NIB	60.6%	66.1%	57.2%	67.4%	59.2%	60.2%
	FedSage+	59.8%	64.5%	57.4%	67.3%	60.1%	60.9%
	FedGCN	59.6%	65.0%	56.9%	65.9%	60.6%	61.3%
	SOS	**49.8%**	**49.8%**	**55.8%**	**53.7%**	**49.7%**	**49.8%**
G_2	SOS-NIB	61.8%	65.6%	56.0%	68.4%	57.3%	61.8%
	FedSage+	61.3%	66.8%	55.9%	69.8%	56.9%	61.4%
	FedGCN	62.1%	67.1%	55.3%	68.8%	56.9%	61.2%
	SOS	**53.4%**	**51.6%**	**54.7%**	**52.4%**	**54.0%**	**52.2%**
G_3	SOS-NIB	53.6%	57.3%	61.3%	65.5%	55.4%	59.5%
	FedSage+	51.8%	54.7%	61.2%	65.5%	55.3%	60.6%
	FedGCN	51.7%	54.1%	60.8%	64.5%	55.6%	61.0%
	SOS	**41.5%**	**46.1%**	**51.9%**	**50.6%**	**51.9%**	**50.6%**
G_4	SOS-NIB	64.3%	70.4%	61.6%	66.3%	58.4%	61.6%
	FedSage+	62.7%	68.9%	62.0%	68.2%	58.9%	62.7%
	FedGCN	62.6%	67.2%	61.5%	67.3%	58.4%	62.5%
	SOS	**50.7%**	**50.2%**	**53.3%**	**51.7%**	**49.3%**	**49.6%**

Additionally, since we apply the proposed approach in federated scenarios, two concerns pop up. 1) *Is the new dataset compatible with the original GNN model due to the change in dataset size?* 2) *Will the down-sampling process on the local graph datasets affect the overall system training effect?* For the first concern, as mentioned in Section 3.1, we treat the learning process of GNN models in an inductive way, therefore, the change in graph size will not cause any incompatibility problem. For the second concern, we would like to mention that the paradigm of the proposed approach is akin to dropout-related FL approaches [3, 19], where some training elements (e.g., participated clients, neuron links, or model weights) are dropped out in the training procedure. The difference is that the majority of dropout-related FL approaches such as Federated Dropout [3] select a subset of *shared model* to locally train and update while we select a subset (IB-subgraph) of the *local dataset* (original subgraph) to locally train the shared model. On the one hand, the performance of federated aggregation algorithms (e.g., FedAvg) has been demonstrated to be robust to even benefit from these dropout operations [3]. On the other hand, compared with the change in model weights or participated clients, which directly influence the model aggregation at the server side, the change in local training data samples will have less influence on the model aggregation. Our scheme is similar to dropping out some local training data samples in the FL systems on regular data (e.g., images). While the data samples in node classification tasks, i.e., nodes, may strongly correlate to each other due to the existence of edges, the IB principle enables the proposed Sub-GIB method can preserve useful edges in the process of preserving task-relevant information. Recall Eq. 1, once local models can effectively learn predictive information from these IB-subgraphs by optimizing local empirical risk $\mathcal{R}_i$, model aggregation algorithms are capable of handling these model updates to develop a generalized and accurate global model [16, 48]. Meanwhile, one limitation is that the influence of non-IIDness between different IB-subgraphs on the FGL performance is not specifically investigated in this work, which can be a possible direction for future work.

Table 3: Prediction Accuracy Comparison of Overall Federated Learning System.

# Client		Cora	Citeseer	PubMed
$k = 2$	SOS-NIB	79.3%	79.9%	80.8%
	FedSage+	84.2%	85.7%	86.6%
	FedGCN	73.0%	61.7%	78.0%
	SOS-HCW	79.1%	82.0%	81.3%
	SOS	78.6%	78.6%	81.3%
$k = 4$	SOS-NIB	84.8%	82.2%	75.0%
	FedSage+	85.4%	86.2%	82.6%
	FedGCN	80.3%	70.6%	72.4%
	SOS-HCW	83.7%	81.8%	73.3%
	SOS	84.2%	81.6%	74.7%
$k = 8$	SOS-NIB	81.6%	74.7%	84.3%
	FedSage+	85.4%	73.57%	86.2%
	FedGCN	76.2%	66.6%	74.8%
	SOS-HCW	76.4%	80.1%	81.4%
	SOS	78.4%	74.4%	84.2%

Further justification of our proposed approach in terms of privacy and utility can be referred to as the experimental results in Sections 5.2 and 5.3.

5 EXPERIMENTAL EVALUATION

5.1 Experiment Preparation

In our experiments, we employ three real-world and widely-adopted graph-structured datasets, namely, Cora [27], Citeseer [27], and PubMed [22]. The statistical information of the three datasets is summarized in Table 1. Following the previous work [49], the ratios of training, validating, and testing sets are 60%, 20%, and 20%, respectively.

To simulate the non-IID characteristics of graph data distributed across different clients, we adopt Metis partitioner [12] to partition the global graph into K subgraphs for corresponding K clents, and further construct their datasets. Metis can ensure a balanced distribution of nodes to different subgraphs. For example, the numbers of nodes of the four partitioned subgraphs of Cora are 696, 661, 688, and 663.

Unless otherwise stated, for the FGL system, the number of clients is set to $K = 4$. We set the global training epoch $T = 50$. The number of local training epochs is set to 10 for Cora and Citeseer and 2 for PubMed. The local GNN models are optimized by Adam with a learning rate $\eta = 0.001$. For SOS, the hyperparameters are set as: $\beta = 0.2$, $\rho = 0.5$, and $\gamma = 0.5$ — the influence of different settings will be investigated later. The epoch of IB optimization is set as $T_{\text{IB}} = 150$. As one of the most investigated models in the graph learning domain, graph convolutional networks (GCN) [13] is adopted as the GNN model to be trained in the FGL system. We employ a 2-layer GCN with the hidden space size of 16 as did in most existing works. Particularly, for the GCN used in Sub-GIB, the hidden space size (i.e., H in Figure 4) is set to 512.

As we are the first to investigate the novel yet significant scenario — subgraph-level FGL against MIA attacks, there are no completely corresponding baselines from existing works. Nevertheless, we consider that two state-of-the-art FGL methods, **FedSage** [49] and **FedGCN** [44], could be good performance benchmarks. In particular, according to the referenced literature, we adopt FedSage+, which is the best-performance one of FedSage. To provide a fair comparison, the setting of FedSage+ is adjusted according to our investigated scenario. Additionally, we first introduce two derived approaches of SOS as the baselines, namely, 1) **SOS-NIB**: No IB-Subgraph, equal to the vanilla FGL as introduced in Section 3.1, which just uses the original subgraphs for local training. 2) **SOS-HCW**: High Card Win, different from the rank-based one in Section 4.3, the larger of scores IN and OUT decides the retention of the node, that is: reserve the node if it has IN ≥ OUT and discarded otherwise.

SOS and the baseline approaches are implemented with PyTorch using half-precision (i.e., float16). All experiments are conducted on a computing server with two Intel Xeon E5 CPUs, and eight nVidia GTX 2080 Ti GPUs are employed for neural networks' computing acceleration. To alleviate the randomness, experiments for each setting are run over five repetitions.

5.2 Robustness to Model Inversion Attack

To validate the effectiveness of our proposed scheme defense against MIA, we conduct a case study of MIA against FGL following the one described in Section 3.2. Particularly, we introduce a state-of-the-art MIA approach, GraphMI [54], which is utilized by the central server to attack a targeted client. GraphMI integrates projected gradient descent and graph autoencoder, which can be regarded as a very threatening MIA adversary. The setting of GraphMI (including architecture and parameters) follows the recommended one in the original literature with minuscule non-algorithmic changes in the practical implementation. As did in [54] and [9], we use two metrics to evaluate the attack, namely, area under the ROC curve (AUC) and average precision (AP) — the larger the AUC or AP, the more successful the MIA. Furthermore, as the subgraph structures vary from client, we in particular show the performance of MIA on different clients to provide a more comprehensive demonstration.

From the results shown in Table 2, we can find that the AUC and AP of MIA on SOS is significantly lower than those on SOS-NIB. When SOS is adopted, the results of AUC are suppressed to 50%, which means that the MIA model is almost making random guesses in this case. That is to say that the threat of MIA is greatly reduced by SOS. The results highlight the advantage of the proposed scheme: as the IB-subgraph developed by SOS is considerably distorted from the original subgraph, less information regarding the original subgraph can be learned by a GNN model. This condition further prevents MIA from reasoning relevant information about the structure of the original subgraph from the model weights. Additionally, it can be observed that for different clients, the robustness brought by the IB-subgraphs (i.e., the downgrade of MIA performance) shows differently. This phenomenon can be explained by that GraphMI predicts the graph structure by a graph autoencoder, where the neural network nature renders randomness in the prediction process. Additionally, we can find the MIA accuracy on FedSage+ is close to that on vanilla FGL and even higher in some cases. This may be due to the missing neighbor mechanism of FedSage+ making the local GNN model learn more information

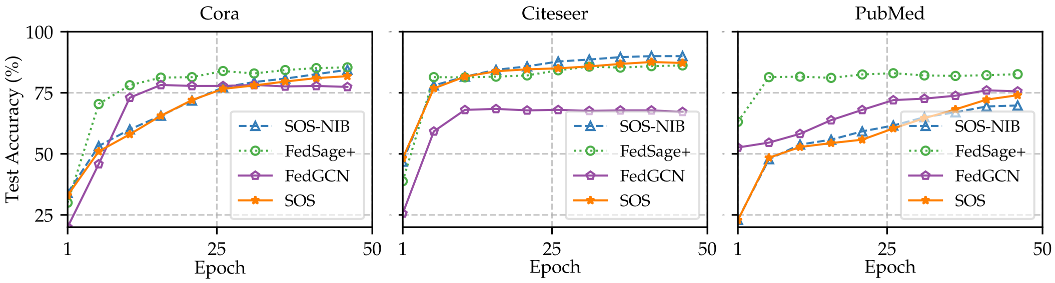

Figure 5: Comparison of training curves with $K = 4$.

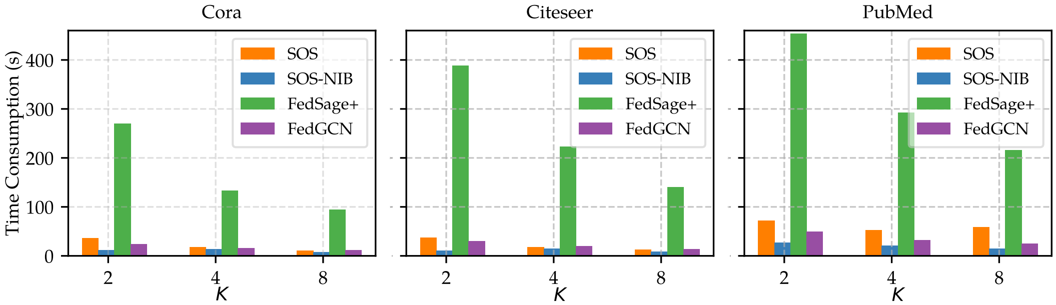

Figure 6: Comparison of training time consumption.

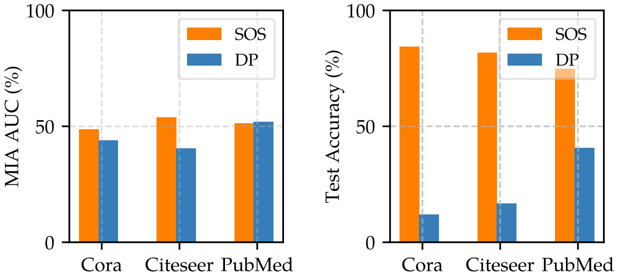

Figure 7: Comparison of MIA resistance and prediction accuracy between LDP and the proposed approach.

about a single subgraph, which makes the updated model updates contain more potential information that MIA can use. The accuracy of MIA on FedGCN is closing to the two, slightly better in some cases. Overall, SOS considerably outperforms the two FGL benchmarks on MIA resistance.

5.3 Accuracy of Federated Graph Learning

As FL systems endeavor to train an accurate global model, the prediction accuracy of the global model is a crucial metric for the goodness of a FL system. We first evaluate the developed global GNN model's prediction accuracy on the testing set, and compare that between SOS and baselines.

The results are presented in Table 3 and Figure 5. We observe that SOS develops a very close accuracy performance to the SOS-NIB, even though the IB-subgraph developed by SOS is smaller and has less information than the original subgraph. The result demonstrates that the developed IB-subgraph can provide the GNN model with sufficient predictive information, which further justifies the effectiveness of our Sub-GIB approach: an informative subgraph in terms of prediction can be extracted. The variant SOS-HCW can also obtain a satisfactory prediction accuracy; however, the no-node-reserved result as concerned in Section 4.3 appears according to our offline tests. In this aspect, the stability of SOS is greater than SOS-HCW credited to the protection mechanism provided by the Top-K algorithm. While FedSage+ demonstrates a remarkable prediction accuracy, the success is on the premise of sacrificing computational efficiency as the involved missing link prediction is time-costly. As shown in Figure 6, the training time consumption of FedSage+ can be ten times as much as that of SOS. On the other hand, we can find that the proposed one is time-efficient from the results due to the one-off generation mechanism of IB-subgraphs. For FedGCN, its training time consumption is satisfying thanks to its communication-optimized mechanism. However, its performance is not robust (refer to the much lower accuracy on Citeseer).

Additionally, we compare prediction accuracy under FGL settings with different numbers of clients, i.e., $K = 2, 4, 8$. Generally, the results on $K = 2, 4, 8$ demonstrate a similar pattern. The only exception occurs when we set $K = 8$ on Cora, where there is an accuracy drop from 81.6% to 78.4%; however, this drop is within the acceptable range.

5.4 Comparison with Differential Privacy-based Defenses

Local differential privacy (LDP) [55] is a widely-adopted technique for protecting local data privacy in FL systems, where a common way of achieving LDP in FL systems is the perturbation of local model updates by calibrated noise. As introduced in Section **??**,

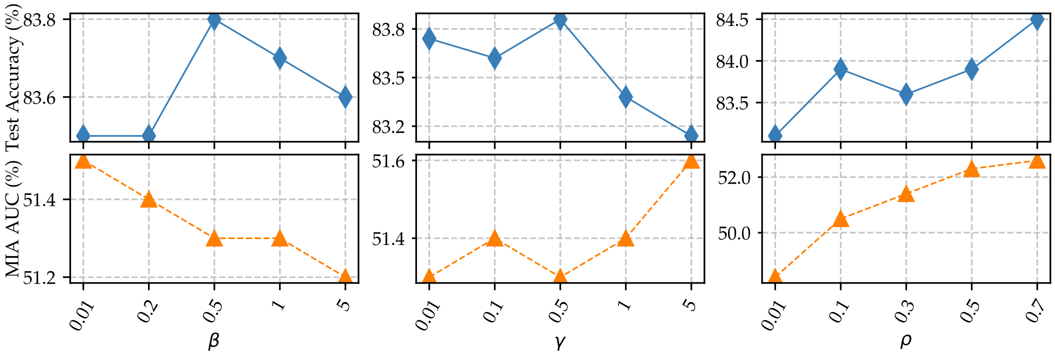

Figure 8: The sensitivity of the proposed scheme to hyperparameter $\beta \in 0.01, 0.2, 0.5, 1, 5$, $\gamma \in 0.01, 0.1, 0.5, 1, 5$, and $\rho \in 0.01, 0.1, 0.3, 0.5, 0.7$ on Cora. AUC is obtained by averaging the MIA results of all four clients.

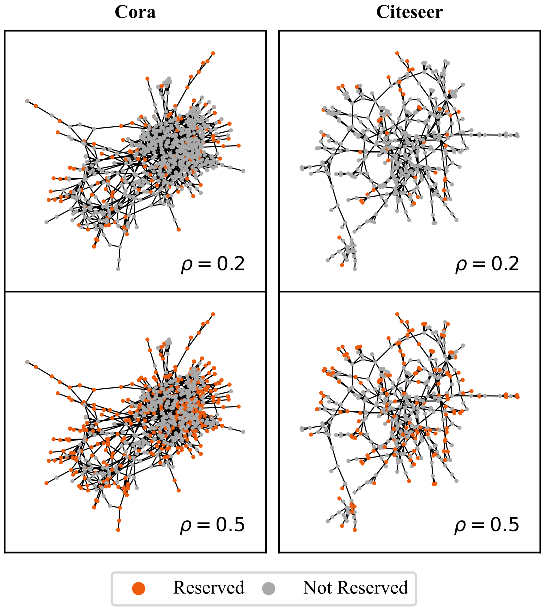

Figure 9: Visualization of reserved nodes in IB-subgraphs. Note that the singletons are presented in this visualization but discarded in practice.

we consider MIA against FGL system, which utilizes the gradient information from the model updates. Amongst various DP-based paradigms in the context of FL, LDP on model updates would be more effective in defending against such attacks [17]. Therefore, we hereby compare the performance between LDP and the proposed approach. Particularly, we empirically adopt the Laplacian mechanism for LDP and add noise from Laplacian distribution $\mathrm{Lap}(0, \frac{s}{\epsilon})$ onto the local GNN updates before aggregation. The sensitivity parameter is set as $s = 1$ as a matter of experience. Moreover, our offline fine-tuning suggests that we can obtain a set of noise enabling a comparable MIA resistance with the proposed scheme when the exponential decay ϵ is set to 1.

We present the comparison of prediction accuracy and MIA resistance between LDP and the proposed scheme under the aforementioned settings in Figure 7. We can clearly observe that the prediction accuracy of FGL with LDP deteriorates considerably under such a noise level. This phenomenon implies a huge utility drop on the GNN models' updates when applying such noises. We can conclude that the proposed approach can achieve a more satisfactory tradeoff between data utility and privacy in the investigated scenario of FGL systems, compared with LDP.

5.5 Sensitivity Studies of Hyperparameters

The proposed scheme mainly incorporates three hyperparameters to control the optimization process, namely, the Lagrange multiplier that controls the distortion extent of $G_i^{\mathrm{IB}} - \beta$, the Lagrange multiplier that controls the tradeoff of contribution between the holistic feature and regional feature $- \gamma$, and the ratio controlling the lowest number of the nodes reserved in $G_i^{\mathrm{IB}} - \rho$. They play a pivotal role in determining the finally generated IB-subgraphs.

We can draw several conclusions from the results shown in Figure 8. The larger the ρ, the higher the AUC of MIA and the lower the accuracy. This is due to that the larger ρ makes more nodes in the original subgraph reserved in the IB-subgraph, making the GNN model can learn sufficient node interaction and develop more accurate predictions. Conversely, the information recorded in the GNN model can also leave exploitable loopholes for MIA. This is the reason why the AUC of MIA is higher when there are more nodes reserved in the IB-subgraphs. For β, a larger one means the MI with the original subgraph will be less considered in the developed IB-subgraph. This makes the developed IB-subgraph reserve less information about the original subgraph. Therefore, the larger β can render a lower AUC of MIA. In terms of γ, a larger one will take less influence of regional feature into account when doing MI estimation; the results show that this will degenerate the prediction performance. This implies the significance of regional features in the Sub-GIB optimization, which will lead to IB-subgraph containing more predictive features regarding structure.

Generally, SOS is more sensitive to the change of ρ as it directly controls the scale of finally generated IB-subgraphs while β and γ mainly control the optimization of Sub-GIB. While the sensitivity of SOS to the change of β and γ is less remarkable, appropriate fine-tuning can contribute to developing more reasonable IB subgraphs to affect the final performance.

In addition, the results of generated IB-subgraphs are visualized in Figure 9. We can observe that the topology among reserved nodes remains the backbone of that in the original subgraph. Furthermore, when ρ becomes larger (from 0.2 to 0.5), the newly included reserved nodes will be uniformly distributed around the existing reserved nodes. A preliminary hypothesis is that the predictability (utility) and privacy of the IB-subgraph is closely related to such a pattern. More in-depth analysis of this phenomenon will be conducted in future work

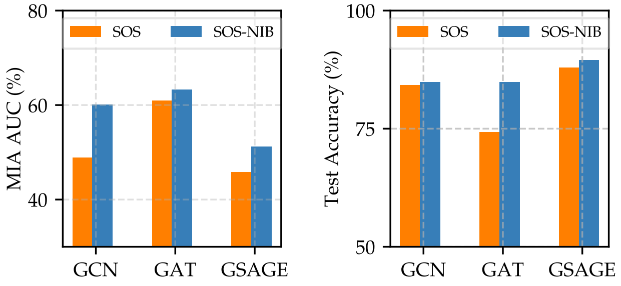

Figure 10: Comparison of MIA resistance and prediction accuracy on different GNNs.

5.6 Generalization Ability on Different GNNs

In Figure 10, we present the performance of SOS on different GNNs. In addition to the default GCN, we introduce graph attention network (GAT) [33] and GraphSAGE [7] in this case study. Generally, we observe that SOS performs better on GCN and GSAGE than GAT. This can be explained by the fact that the attention mechanism in GAT can help establish a stronger correlation between the graph structure and node labels, which can benefit the MIA's performance. Meanwhile, GAT may also be more susceptible to structural changes in the graph caused by SOS, which could lead to a decrease in prediction accuracy.

6 SUMMARY AND FUTURE WORK

In this paper, we propose a novel scheme named SOS for federated graph learning, which can defend against model inversion attacks effectively. Our proposed scheme mainly leverages the information bottleneck principle to identify a smaller IB-subgraph from the original subgraph of each client and feed these IB-subgraphs into FGL systems for local GNN model's training. Particularly, we use neural networks to achieve the mutual information estimation between original subgraphs and IB-subgraphs. We also devise an IB-subgraph generation algorithm to develop reasonable IB-subgraphs. Comprehensive experiments on three datasets demonstrate that the proposed scheme can significantly improve the robustness of FGL to model inversion attacks. Meanwhile, the prediction accuracy of the global GNN model trained on smaller IB-subgraphs is at the same level as the one trained on the original subgraphs.

In our future work, we will conduct a theoretical analysis of the privacy guarantee provided by the IB-subgraph. In addition, it is interesting to explore our scheme with further MIA approaches in various FGL attack settings.

ACKNOWLEDGMENTS

This research is supported by the Australian Research Council (ARC) LP190100676.

REFERENCES

[1] Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. 2017. Deep Variational Information Bottleneck. In *Proc. International Conference on Learning Representations*.

[2] Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. 2017. QSGD: Communication-efficient SGD via gradient quantization and encoding. *Advances in neural information processing systems* 30 (2017).

[3] Sebastian Caldas, Jakub Konečny, H Brendan McMahan, and Ameet Talwalkar. 2018. Expanding the reach of federated learning by reducing client resource requirements. *arXiv preprint arXiv:1812.07210* (2018).

[4] Enyan Dai, Tianxiang Zhao, Huaisheng Zhu, Junjie Xu, Zhimeng Guo, Hui Liu, Jiliang Tang, and Suhang Wang. 2022. A Comprehensive Survey on Trustworthy Graph Neural Networks: Privacy, Robustness, Fairness, and Explainability. *arXiv preprint arXiv:2204.08570* (2022).

[5] Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. 2014. Privacy in pharmacogenetics: An {End-to-End} case study of personalized warfarin dosing. In *23rd USENIX Security Symposium (USENIX Security 14)*. 17–32.

[6] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In *Proc. International Conference on Learning Representations*.

[7] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Proc. Advances in neural information processing systems* 30 (2017).

[8] Chaoyang He, Emir Ceyani, Keshav Balasubramanian, Murali Annavaram, and Salman Avestimehr. 2021. Spreadgnn: Serverless multi-task federated learning for graph neural networks. *arXiv preprint arXiv:2106.02743* (2021).

[9] Xinlei He, Jinyuan Jia, Michael Backes, Neil Zhenqiang Gong, and Yang Zhang. 2021. Stealing links from graph neural networks. In *30th USENIX Security Symposium (USENIX Security 21)*. 2669–2686.

[10] Briland Hitaj, Giuseppe Ateniese, and Fernando Perez-Cruz. 2017. Deep models under the GAN: information leakage from collaborative deep learning. In *Proc. ACM SIGSAC conference on computer and communications security*. 603–618.

[11] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. 2018. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*.

[12] George Karypis and Vipin Kumar. 1997. *METIS: A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices.* Technical Report. Computer Science & Engineering (CS&E) Technical Reports.

[13] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *Proc. International Conference on Learning Representations*.

[14] John D Kraemer, Andrew A Strasser, Eric N Lindblom, Raymond S Niaura, and Darren Mays. 2017. Crowdsourced data collection for public health: A comparison with nationally representative, population tobacco use data. *Preventive Medicine* 102 (2017), 93–99.

[15] Anusha Lalitha, Osman Cihan Kilinc, Tara Javidi, and Farinaz Koushanfar. 2019. Peer-to-peer federated learning on graphs. *arXiv preprint arXiv:1901.11173* (2019).

[16] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. 2019. On the Convergence of FedAvg on Non-IID Data. In *International Conference on Learning Representations*.

[17] Lingjuan Lyu, Han Yu, Xingjun Ma, Chen Chen, Lichao Sun, Jun Zhao, Qiang Yang, and S Yu Philip. 2022. Privacy and robustness in federated learning: Attacks and defenses. *IEEE transactions on neural networks and learning systems* (2022).

[18] Ali Makhdoumi, Salman Salamatian, Nadia Fawaz, and Muriel Médard. 2014. From the information bottleneck to the privacy funnel. In *2014 IEEE Information Theory Workshop (ITW 2014)*. IEEE, 501–505.

[19] Yuzhu Mao, Zihao Zhao, Guangfeng Yan, Yang Liu, Tian Lan, Linqi Song, and Wenbo Ding. 2022. Communication-efficient federated learning with adaptive quantization. *ACM Transactions on Intelligent Systems and Technology (TIST)* 13, 4 (2022), 1–26.

[20] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*. PMLR, 1273–1282.

[21] Tamara T Mueller, Dmitrii Usynin, Johannes C Paetzold, Daniel Rueckert, and Georgios Kaissis. 2022. SoK: Differential Privacy on Graph-Structured Data. *arXiv preprint arXiv:2203.09205* (2022).

[22] Galileo Namata, Ben London, Lise Getoor, Bert Huang, and U Edu. 2012. Query-driven active surveying for collective classification. In *10th International Workshop on Mining and Learning with Graphs*, Vol. 8. 1.

[23] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. 2016. f-gan: Training generative neural samplers using variational divergence minimization. *Advances in neural information processing systems* 29 (2016).

[24] Zhen Peng, Wenbing Huang, Minnan Luo, Qinghua Zheng, Yu Rong, Tingyang Xu, and Junzhou Huang. 2020. Graph representation learning via graphical mutual information maximization. In *Proc. The Web Conference 2020*. 259–270.

[25] Mohamed Sabt, Mohammed Achemlal, and Abdelmadjid Bouabdallah. 2015. Trusted execution environment: what it is, and what it is not. In *2015 IEEE Trustcom/BigDataSE/ISPA*, Vol. 1. IEEE, 57–64.

[26] Lalitha Sankar, S Raj Rajagopalan, and H Vincent Poor. 2013. Utility-privacy tradeoffs in databases: An information-theoretic approach. *IEEE Transactions on Information Forensics and Security* 8, 6 (2013), 838–852.

[27] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. 2008. Collective classification in network data. *AI magazine* 29, 3 (2008), 93–93.

[28] Fan-Yun Sun, Jordon Hoffman, Vikas Verma, and Jian Tang. 2020. InfoGraph: Unsupervised and Semi-supervised Graph-Level Representation Learning via Mutual Information Maximization. In *International Conference on Learning Representations*.

[29] Lichao Sun, Yingtong Dou, Carl Yang, Ji Wang, Philip S Yu, Lifang He, and Bo Li. 2018. Adversarial attack and defense on graph data: A survey. *arXiv preprint arXiv:1812.10528* (2018).

[30] Qingyun Sun, Jianxin Li, Hao Peng, Jia Wu, Xingcheng Fu, Cheng Ji, and S Yu Philip. 2022. Graph structure learning with variational information bottleneck. In *Proc. AAAI Conference on Artificial Intelligence*, Vol. 36. 4165–4174.

[31] Naftali Tishby, Fernando C. Pereira, and William Bialek. 1999. The information bottleneck method. In *Proc. of the 37-th Annual Allerton Conference on Communication, Control and Computing*. 368–377. https://arxiv.org/abs/physics/0004057

[32] Bo-Wei Tseng and Pei-Yuan Wu. 2020. Compressive privacy generative adversarial network. *IEEE Transactions on Information Forensics and Security* 15 (2020), 2499–2513.

[33] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *Proc. International Conference on Learning Representations*.

[34] Petar Velickovic, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R. Devon Hjelm. 2019. Deep Graph Infomax. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. https://openreview.net/forum?id=rklz9iAcKQ

[35] Binghui Wang, Ang Li, Hai Li, and Yiran Chen. 2020. Graphfl: A federated learning framework for semi-supervised node classification on graphs. *arXiv preprint arXiv:2012.04187* (2020).

[36] Rundong Wang, Xu He, Runsheng Yu, Wei Qiu, Bo An, and Zinovi Rabinovich. 2020. Learning efficient multi-agent communication: An information bottleneck approach. In *International Conference on Machine Learning*. PMLR, 9908–9918.

[37] Zhibo Wang, Mengkai Song, Zhifei Zhang, Yang Song, Qian Wang, and Hairong Qi. 2019. Beyond inferring class representatives: User-level privacy leakage from federated learning. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2512–2520.

[38] Bang Wu, Xiangwen Yang, Shirui Pan, and Xingliang Yuan. 2021. Adapting membership inference attacks to GNN for graph classification: approaches and implications. In *2021 IEEE International Conference on Data Mining (ICDM)*. IEEE, 1421–1426.

[39] Bang Wu, Xiangwen Yang, Shirui Pan, and Xingliang Yuan. 2022. Model extraction attacks on graph neural networks: Taxonomy and realisation. In *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*. 337–350.

[40] Chuhan Wu, Fangzhao Wu, Yang Cao, Yongfeng Huang, and Xing Xie. 2021. Fedgnn: Federated graph neural network for privacy-preserving recommendation. *arXiv preprint arXiv:2102.04925* (2021).

[41] Huijun Wu, Chen Wang, Yuriy Tyshetskiy, Andrew Docherty, Kai Lu, and Liming Zhu. 2019. Adversarial examples for graph data: deep insights into attack and defense. In *Proc. International Joint Conference on Artificial Intelligence*. 4816–4823.

[42] Tailin Wu, Hongyu Ren, Pan Li, and Jure Leskovec. 2020. Graph information bottleneck. *Advances in Neural Information Processing Systems* 33 (2020), 20437–20448.

[43] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)* 10, 2 (2019), 1–19.

[44] Yuhang Yao and Carlee Joe-Wong. 2022. Fedgcn: Convergence and communication tradeoffs in federated training of graph convolutional networks. *arXiv preprint arXiv:2201.12433* (2022).

[45] Junchi Yu, Tingyang Xu, Yu Rong, Yatao Bian, Junzhou Huang, and Ran He. 2020. Graph Information Bottleneck for Subgraph Recognition. In *International Conference on Learning Representations*.

[46] Junchi Yu, Tingyang Xu, Yu Rong, Yatao Bian, Junzhou Huang, and Ran He. 2021. Recognizing predictive substructures with subgraph information bottleneck. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

[47] He Zhang, Bang Wu, Xingliang Yuan, Shirui Pan, Hanghang Tong, and Jian Pei. 2022. Trustworthy graph neural networks: Aspects, methods and trends. *arXiv preprint arXiv:2205.07424* (2022).

[48] Jian Zhang, Christopher De Sa, Ioannis Mitliagkas, and Christopher Ré. 2016. Parallel SGD: When does averaging help? *arXiv preprint arXiv:1606.07365* (2016).

[49] Ke Zhang, Carl Yang, Xiaoxiao Li, Lichao Sun, and Siu Ming Yiu. 2021. Subgraph federated learning with missing neighbor generation. *Advances in Neural Information Processing Systems* 34 (2021).

[50] Qi Zhang, Jianlong Chang, Gaofeng Meng, Shibiao Xu, Shiming Xiang, and Chunhong Pan. 2019. Learning graph structure via graph convolutional networks. *Pattern Recognition* 95 (2019), 308–318.

[51] Zhikun Zhang, Min Chen, Michael Backes, Yun Shen, and Yang Zhang. 2022. Inference attacks against graph neural networks. In *Proc. USENIX Security Symposium*. 1–18.

[52] Zaixi Zhang, Jinyuan Jia, Binghui Wang, and Neil Zhenqiang Gong. 2021. Backdoor attacks to graph neural networks. In *Proc. ACM Symposium on Access Control Models and Technologies*. 15–26.

[53] Zaixi Zhang, Qi Liu, Zhenya Huang, Hao Wang, Chee-Kong Lee, and Enhong Chen. 2022. Model inversion attacks against graph neural networks. *IEEE Transactions on Knowledge and Data Engineering* (2022).

[54] Zaixi Zhang, Qi Liu, Zhenya Huang, Hao Wang, Chengqiang Lu, Chuanren Liu, and Enhong Chen. 2021. GraphMI: Extracting Private Graph Data from Graph Neural Networks. In *Proc. International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, Zhi-Hua Zhou (Ed.). ijcai.org, 3749–3755.

[55] Yang Zhao, Jun Zhao, Mengmeng Yang, Teng Wang, Ning Wang, Lingjuan Lyu, Dusit Niyato, and Kwok-Yan Lam. 2020. Local differential privacy-based federated learning for internet of things. *IEEE Internet of Things Journal* 8, 11 (2020), 8836–8853.

[56] Longfei Zheng, Jun Zhou, Chaochao Chen, Bingzhe Wu, Li Wang, and Benyu Zhang. 2021. Asfgnn: Automated separated-federated graph neural network. *Peer-to-Peer Networking and Applications* 14, 3 (2021), 1692–1704.

[57] Daniel Zügner, Amir Akbarnejad, and Stephan Günnemann. 2018. Adversarial attacks on neural networks for graph data. In *Proc. ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2847–2856.