Federated Graph Condensation with Information Bottleneck Principles

Bo Yan[1,2], Sihao He[1], Cheng Yang[1], Shang Liu[3], Yang Cao[2], Chuan Shi[1]*

[1]Beijing University of Posts and Telecommunications
[2]Institute of Science Tokyo
[3]China University of Mining and Technology
{boyan, sihaohe, yangcheng, shichuan}@bupt.edu.cn, shang@cumt.edu.cn, cao@c.titech.ac.jp

Abstract

Graph condensation (GC), which reduces the size of a large-scale graph by synthesizing a small-scale condensed graph as its substitution, has benefited various graph learning tasks. However, existing GC methods rely on centralized data storage, which is unfeasible for real-world decentralized data distribution, and overlook data holders' privacy-preserving requirements. To bridge this gap, we propose and study the novel problem of federated graph condensation (FGC) for graph neural networks (GNNs). Specifically, we first propose a general framework for FGC, where we decouple the typical gradient matching process for GC into client-side gradient calculation and server-side gradient matching, integrating knowledge from multiple clients' subgraphs into one smaller condensed graph. Nevertheless, our empirical studies show that under the federated setting, the condensed graph will consistently leak data membership privacy, i.e., the condensed graph during federated training can be utilized to steal training data under the membership inference attack (MIA). To tackle this issue, we innovatively incorporate information bottleneck principles into the FGC, which only needs to extract partial node features in one local pre-training step and utilize the features during federated training. Theoretical and experimental analyses demonstrate that our framework consistently protects membership privacy during training. Meanwhile, it can achieve comparable and even superior performance against existing centralized GC and federated graph learning (FGL) methods.

1 Introduction

Graph data, such as social networks and transportation networks, is ubiquitous in the real world. Graph neural networks (GNNs) have been prevalent for modeling these graphs and achieved remarkable success in recent years (Hamilton, Ying, and Leskovec 2017; Kipf and Welling 2017; Velickovic et al. 2018). However, GNNs have struggled with real-world large-scale graphs due to memory and computational overheads, which are further exacerbated when the models need to be trained multiple times, e.g., neural architecture search (NAS) (Zhang et al. 2022) and continual learning (Li and Hoiem 2016). A natural solution from the data-centric view is graph condensation (GC) (Jin et al. 2022b; Liu et al. 2022; Zheng et al. 2023; Liu, Bo, and Shi 2024), as shown in Figure 1(a), which reduces large-scale graphs into small ones but keeps the utility of original graphs, facilitating the storage of graph data and computations of downstream tasks. As a typical technique of GC, gradient matching (Jin et al. 2022b,a) forces the gradients of GNN models trained on the original graph and condensed graph to be the same, maintaining a comparable model performance, but largely reducing the graph size.

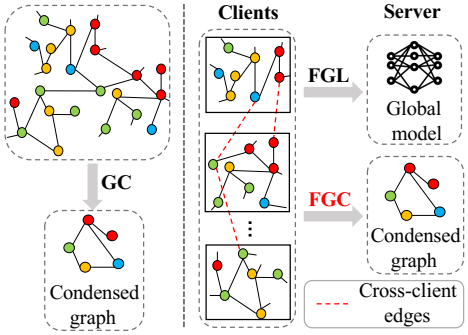

Figure 1: The comparison between (a) centralized graph condensation (GC), (b) federated graph learning (FGL) and federated graph condensation (FGC).

Despite significant progress in GC, existing works all hold a basic assumption that the graph data is centrally stored. In reality, the whole graph is divided into multiple subgraphs held by different parties, as in healthcare systems, a hospital only possesses a subset of patients (nodes), their information (attributes), and interactions (links). Due to privacy concerns, data holders are unwilling to share their data and many strict regulations such as GDPR (Voigt and Von dem Bussche 2017) also forbid collecting data arbitrarily, which makes centralized GC unfeasible. Federated graph learning (FGL) (Yao et al. 2023; Wan et al. 2022; Zhang et al. 2021) has emerged in recent years to endow collaboratively training a global model without exposing local data, as is shown in Figure 1(b). Typically, they are dedicated to training a global model by exchanging encrypted information for recovering missing structures (e.g., cross-client edges and neighbors). However, they still suffer from bur-

*Corresponding author.

densome computations of traditional GNNs for large-scale graph training, especially in the cross-silo scenario (Gao, Yao, and Yang 2022). For example, conducting NAS for FGL inevitably requires multiple times of federated training, exacerbating both computation and communication overheads. Therefore, an important yet unexplored area lies in considering GC in the federated setting.

To bridge this gap, we investigate the novel problem of federated graph condensations (FGC), as shown in Figure 1 (b), which aims to collaboratively learn a small graph that contains knowledge from different data sources. Subsequently, all clients can access the condensed graph for downstream applications. This task is non-trivial due to two significant challenges. ***(1) How to preserve the utility of the condensed graph that is learned from distributed subgraphs?*** Unlike centralized GC the graph data is integral, the entire graph is distributed as subgraphs across data holders in FGC. As a result, some critical cross-client information is missing and these subgraphs also suffer from severe heterogeneity, which inevitably degrades the utility of the condensed graph. In this regard, a unified framework for utility-preserving FGC is desired. ***(2) How to preserve the local graph's membership privacy that may be leaked by the condensed graph?*** Membership privacy is widely used to measure information leakage about training data (Nasr, Shokri, and Houmansadr 2019). Unlike local storage of graph data in typical FGL, a condensed graph is released in FGC, which may be utilized to infer the membership privacy of local data using membership inference attack (MIA). Furthermore, the accessibility of the condensed graph enables attackers to train arbitrary models, as shown in Figure 2, rendering traditional model-based defenses against MIA infeasible. Thus, new methods are needed to defend against MIA in FGC.

To tackle these challenges, we propose a Federated Graph Condensation framework (FedGC). (1) To preserve data utility, we propose a general framework for FGC, the core of which is that the condensed graph is learned by matching the aggregated gradients from clients. Particularly, the gradients are aggregated in a class-aware weighted manner to tackle data heterogeneity. Furthermore, an encrypted one-step communication between the server and clients is adopted to recover the cross-client neighbors. During the evaluation phase, the model trained on the condensed graph will be fine-tuned by local data for better personalization. (2) To preserve membership privacy, we further propose a novel local graph transformation module with information bottleneck principles (Alemi et al. 2017). We first empirically reveal that the condensed graph will consistently leak membership privacy. To address this, each subgraph will be transformed locally before condensation by fixing the graph structure unchanged and extracting sufficient but minimal node features. Then the transformed graph will act as a proxy of the original graph to conduct FGC, thereby protecting the local membership privacy. Besides, to alleviate the performance decay in graph transformation caused by scarce labeled nodes and heterogeneous feature space, we propose a one-time self-training strategy to label nodes and utilize a fixed pre-trained shared model to align the feature space. Finally, we give theoretical and empirical analysis to prove the utility- and privacy-preserving properties of FedGC. The major contributions of this paper are summarized as follows:

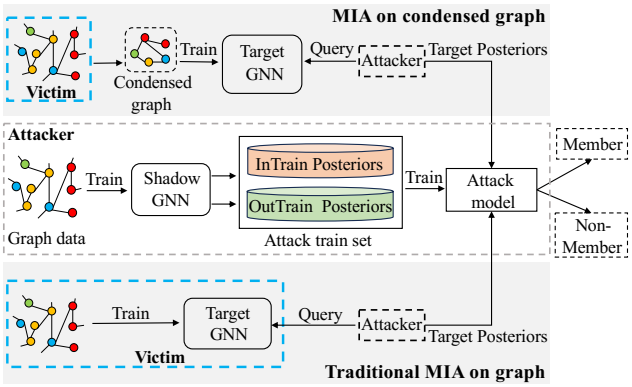

Figure 2: The comparison between MIA on condensed graphs and traditional MIA on graphs. The components circled by blue dashed lines are invisible to attackers.

- To the best of our knowledge, this is the first work to study federated graph condensation (FGC), which is an important and practical task in real-world scenarios.
- We design a FedGC framework for FGC. It learns the condensed graph by matching the weighted gradient aggregation from clients. We reveal that the condensed graph will consistently leak membership privacy and further propose a novel local graph transformation module with information bottleneck principles to protect original graphs. Theoretical analysis proves that FedGC can simultaneously maintain the utility of the original graph and preserve membership privacy.
- We conduct extensive experiments on five real-world datasets and show that FedGC outperforms centralized GC and FGL methods, especially in large-scale datasets. Meanwhile, FedGC can consistently protect membership privacy during the whole federated training process.

2 Preliminary

Notations. Let $G = (A, X, Y)$ denote the original graph with adjacency matrix $A \in \mathbb{R}^{N\times N}$, node feature matrix $X \in \mathbb{R}^{N\times d}$ and node label set $Y \in \{1, \ldots, C\}$ over C classes. Similarly, let $G' = (A', X', Y')$ denote the condensed graph. In the federated setting, the whole graph G is divided into multiple subgraphs $\{G_i\}_{i=1}^{M}$ where $G_i = (A_i, X_i, Y_i)$ stored in the i-th client. Let V denote the whole node set and V_i denote the node set of the i-th client, following (Yao et al. 2023), we assume $V = V_1 \cup \ldots \cup V_M$ and $V_i \cap V_j = \emptyset$, i.e., there are no overlapping nodes across clients. Also, we use E to denote the whole edge set and E_i to denote the edge set of i-th client. For an edge $e_{v,u} \in E$, where $v \in V_i$ and $u \in V_j$, we assume $e_{v,u} \in E_i \cup E_j$, i.e., the cross-client edge $e_{v,u}$ is known to the client i and j.

Information bottleneck (Alemi et al. 2017). Given original data $\mathcal{D}$, information bottleneck (IB) is to optimize Z to capture the minimal sufficient information within $\mathcal{D}$ to predict the label Y. The objective of IB can be written as:

$$\min -\mathrm{I}(Y;Z)+\gamma\mathrm{I}(\mathcal{D};Z), \tag{1}$$

where $\mathrm{I}(;)$ means mutual information (MI). Maximizing the first term aims to preserve the utility of $\mathcal{D}$ and minimizing the second term helps filter out the redundant information of $\mathcal{D}$. γ is the parameter that balances two terms.

GC with gradient matching. Given a graph G, GC aims to condense G into a smaller, synthetic graph $G' = (A', X', Y')$ with $A' \in \mathbb{R}^{N'\times N'}$, node feature matrix $X' \in \mathbb{R}^{N'\times d}$, node label set $Y' \in \{1,\dots,C\}^{N'}$ and $N' \ll N$, meanwhile maintaining the utility of G (i.e., achieving comparable performance with the model trained on G). To achieve this, typical solutions update G' by matching the gradients $\nabla\mathcal{L}$ of models trained on G and G' (Zhao, Mopuri, and Bilen 2021), which can be formalized as:

$$\min_{G'} E_{\theta_0\sim P_\theta}[D(\nabla_\theta\mathcal{L}^{G}_{\theta_0}, \nabla_\theta\mathcal{L}^{G'}_{\theta_0})], \tag{2}$$

where $D(\cdot,\cdot)$ is the distance function (e.g., Euclidean distance) and θ_0 is the initialized model parameter.

MIA on condensed graphs. Following (Dong, Zhao, and Lyu 2022), we assume an honest-but-curious server is the strong attacker. Since the condensed graph G' is publicly accessed, the target models can be arbitrarily selected and trained by attacks, which is different from traditional MIA on graphs (the attacker can only query target models (Olatunji, Nejdl, and Khosla 2021)), as shown in Figure 2. Therefore, traditional model-based defenses against MIA are infeasible. Following (Olatunji, Nejdl, and Khosla 2021), we also assume the attacker knows the target data distribution and can access a part of in-distribution data G_s (shadow data). To conduct MIA in FGC scenarios, the attacker first utilizes G_s to train a shadow GNN $\mathcal{S}$. Then, the posterior probabilities of nodes from $\mathcal{S}$ are used to train an attack model $\mathcal{A}$ to distinguish between non-member and member nodes. Meanwhile, the attacker utilizes G' to train a target GNN $\mathcal{T}$. During the attack phase, given a target node $v \in V$ and its L-hop neighbors, $\mathcal{A}$ can infer the membership of v based on its posterior probability obtained by $\mathcal{T}$. Note that the L-hop neighbors of v known to the attacker may not be the exact neighbors, which is also a realistic setting.

Task formulation. Given multiple subgraphs $\{G_i = (A_i, X_i, Y_i)\}_{i=1}^{M}$, each of which stores in the i-th client, FGC aims to condense $\{G_i\}_{i=1}^{M}$ into a smaller graph G' stored in the server. G' should maintain the utilities of original graphs $\{G_i\}_{i=1}^{M}$ (i.e., the model trained on G' should obtain the comparable performance to that of a model trained on the original graphs), meanwhile containing less membership privacy of original graphs.

3 Methodology

In this section, we give a detailed introduction to the proposed model FedGC. We first present a general framework for FGC. Then, we empirically reveal that the condensed graph by the framework is vulnerable to MIA. Thus, a local graph transformation with information bottleneck principles is introduced to defend MIA. Finally, we theoretically prove the utility- and privacy-preserving abilities of FedGC.

3.1 General Framework of FGC

We first review the typical centralized graph condensation in Eq.(2). It can not be directly applied to the federated setting since the integral graph $G = (A, X, Y)$ is divided into multiple subgraphs $\{G_j = (A_j, X_j, Y_j)\}_{j=1}^{M}$. A natural solution is that each client j trains a GNN model locally based on G_j and uploads model gradients to the server for aggregation. To mitigate the impact of node label heterogeneity among different clients and facilitate optimization, for each class c, we first sample a subgraph $G_{j,c} = (A_{j,c}, X_{j,c}, Y_{j,c})$ consisting of nodes with label c and their neighbors, then calculate the model gradients on $G_{j,c}$. The server performs weighted gradient aggregation based on the number of nodes $n_j(c)$. Therefore, we can modify the matching loss in Eq.(2) as:

$$\begin{aligned}
&\min_{G'} \frac{1}{t}\sum_{t=0}^{t=T}\sum_{c=0}^{C}[D(\nabla_\theta\mathcal{L}^{G}_{c,\theta_t}, \nabla_\theta\mathcal{L}^{G'}_{c,\theta_t})], \theta_t \sim P_\theta,\\
&\nabla_\theta\mathcal{L}^{G}_{c,\theta_t} = \sum_{j=1}^{j=M}\frac{n_j(c)}{n(c)}\nabla_\theta\mathcal{L}(GNN_{\theta_t}(A_{j,c}, X_{j,c}), Y_{j,c}),\\
&\nabla_\theta\mathcal{L}^{G'}_{\theta_t,c} = \nabla_\theta\mathcal{L}(GNN_{\theta_t}(A'_c, X'_c), Y'_c).
\end{aligned} \tag{3}$$

That is, at each communication round t, the condensed graph can be updated by matching the gradient $\nabla_\theta\mathcal{L}^{G}_{c,\theta_t}$ and $\nabla_\theta\mathcal{L}^{G'}_{c,\theta_t}$ w.r.t. GNN on current condensed graph G'. Note that we leave the updating of the condensed graph on the server side and the client only needs to upload the gradient of the GNN model, which reduces the communication cost and computation overhead of clients.

As illustrated in (Jin et al. 2022b), it's hard to optimize A', X' and Y' jointly. Therefore, we fix the Y' as the same distribution as the original Y during federated training. To obtain the distribution $P(Y)$, the client j will upload the number of each class $n_j(c)$ in its own Y_j before federated training. We also assume that A' is induced from X': $A'_{i,j} = \sigma(\mathrm{MLP}_\Phi(x'_i, x'_j))$, which largely reduces computation cost.

One-step communication. The challenges in FGL lie in broken local structures and the distributional heterogeneity of subgraphs (Yao et al. 2023; Baek et al. 2023), which are further compounded by the utility-preserving property in FGC. Therefore, inspired by (Yao et al. 2023), we leverage homomorphic encryption (Brakerski, Gentry, and Vaikuntanathan 2014) to securely transfer missing neighbor aggregations among clients. This step is performed only once in advance, and the final aggregated node embeddings are used for FGC. Notably, as at most 2-hop neighbors are sufficient in cross-silo FL (Yao et al. 2023), this step introduces little communication cost while improving performance.

Fine-tuning with local graphs. After the condensed graph G' is well trained, it can be distributed to clients for testing. A typical testing flow in traditional GC is training a target model $\mathcal{T}$ with G', then utilizing the model $\mathcal{T}$ to test local data. However, in FGC, the heterogeneity issue across clients hinders the performance of this one-test-all mode since the condensed graph only captures the general knowledge of multiple subgraphs. Therefore, we propose to use local data to fine-tune $\mathcal{T}$ so that it can adapt well to different data distributions.

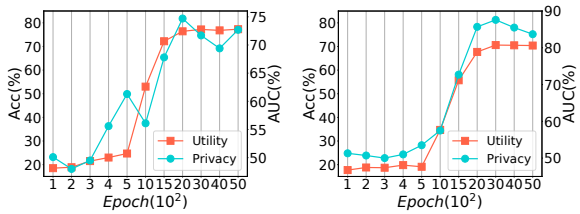

Figure 3: The comparison between the utility of condensed graph (accuracy of node classification ↑) and privacy attack performance (AUC of MIA ↓) during federated training (left: Cora, right: Citeseer).

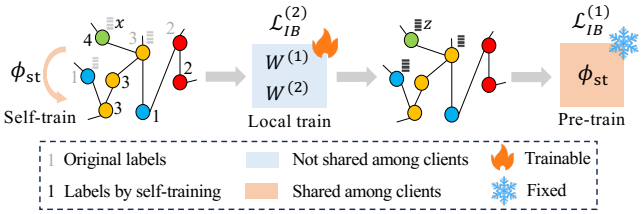

Figure 4: The overall workflow of local graph transformation with information bottleneck principles.

3.2 Local Graph Transformation for Preserving Membership Privacy

Based on the above framework, we can condense multiple large subgraphs into a small graph without exposing local data. However, our empirical study shows that the condensed graph still exposes membership privacy. We perform FGC under the MIA introduced in the preliminary to record the utility and attack performance of the condensed graph. From Fig. 3, we can see that the performance of MIA increases along with the utility, i.e., the condensed graph will gradually leak the membership privacy of original graphs, which intuitively makes sense since it's trained to match the local graphs. Hereby, new mechanisms are desired to condense graphs containing less membership privacy.

To achieve this, we propose to transform the local subgraph into another one and then perform FGC based on these transformed graphs. The new graph should be equipped with two properties: (1) maintaining the utility of the original graph and (2) containing less membership privacy of the original graph. Inspired by recent work (Yang et al. 2023; Dai et al. 2023) that utilizes information bottleneck (IB) principles to extract minimal sufficient features from origin data, we propose to transform original local subgraphs with IB principles. The overall workflow is depicted in Fig. 4. Concretely, for a node in one client, we use $x \in \mathbb{R}^{d\times 1}$ to denote its original features. Then we aim to learn an IB encoder f_x to extract minimal sufficient features z of x: $z = f_x(x, \mathcal{N})$, where $\mathcal{N}$ is the neighborhood of the node. According to the IB principles (Alemi et al. 2017; Wu et al. 2020), our objective function is:

$$\min -\mathrm{I}(z, \mathcal{N}; y) + \gamma \mathrm{I}(z; x, \mathcal{N}). \tag{4}$$

By optimizing Eq.(4), the first term $\mathrm{I}(z, \mathcal{N}; y)$ is maximized, which makes z contain enough information to predict the label y. Meanwhile, the second term $\mathrm{I}(z; x, \mathcal{N})$ is minimized, forcing z to learn less information from x and their neighbors $\mathcal{N}$. As a result, the new feature z can preserve the utility and contains less privacy information of original features x.

Upper bound of the objective function. It's hard to directly optimize Eq.(4) due to the two MI terms. Following the IB principles (Alemi et al. 2017), we derive the upper bound of the two MI terms. First, we introduce a variational distribution $q(y|z, \mathcal{N})$ to approximate $p(y|z, \mathcal{N})$, and we derive the upper bound of $-\mathrm{I}(z, \mathcal{N}; y)$:

$$\begin{aligned} -\mathrm{I}(z, \mathcal{N}; y) &\leq -\mathbb{E}_{p(z,\mathcal{N},y)} \log \frac{q(y|z, \mathcal{N})}{p(y)} \\ &\leq -\mathbb{E}_{p(z,\mathcal{N},y)} \log q(y|z, \mathcal{N}) = \mathcal{L}_{IB}^{(1)}. \end{aligned} \tag{5}$$

Similarly, a variational distribution $q(z)$ is introduced to approximate $p(z)$, then the upper bound of $\mathrm{I}(z; x, \mathcal{N})$ can be calculated by:

$$\begin{aligned} \mathrm{I}(z; x, \mathcal{N}) &\leq \mathbb{E}_{p(z,x,\mathcal{N})} \log \frac{p(z|x, \mathcal{N})}{q(z)} \\ &= \mathbb{E}_{p(x,\mathcal{N})} \mathrm{KL}(p(z|x, \mathcal{N}) || q(z)) = \mathcal{L}_{IB}^{(2)}. \end{aligned} \tag{6}$$

After obtaining the above upper bounds, we get the optimization objective of our local graph transformation:

$$\min \mathcal{L}_{IB} = \mathcal{L}_{IB}^{(1)} + \gamma \mathcal{L}_{IB}^{(2)}. \tag{7}$$

Note that typical graph IB (Wu et al. 2020) also performs graph structure transformation. Considering the joint optimization of features and structures brings burdensome overheads to the clients, we only perform node feature transformation and keep the original structure unchanged, which can be further theoretically proven to defend against MIA.

Instantiating the IB Principle. After giving the above IB principles, we need to instantiate them with different neural network architectures. For $p(z|x, \mathcal{N})$, we assume it follows Gaussian distribution, which is parameterized by MLPs followed by neighbor aggregations. Specifically, we first feed x into a two-layer MLP: $h = W^{(2)}\sigma(W^{(1)}x)$, where $W^{(1)} \in \mathbb{R}^{d\times d}$ and $W^{(2)} \in \mathbb{R}^{2d\times d}$. Then, we perform local neighbor aggregation to obtain hidden representation $\hat{z} = \mathrm{AGGREGATE}(h, \mathcal{N})$. z is sampled from the Gaussian distribution with mean and variance from $\hat{z}$:

$$z \sim N(\mu; \sigma), \quad \mu = \hat{z}[0:d], \quad \sigma = \mathrm{softplus}(\hat{z}[d:2d]). \tag{8}$$

In this process, the reparameterization trick (Kingma and Welling 2014) is applied. We assume $q(z)$ follows the standard normal distribution $\mathcal{N}(0, 1)$, then the $\mathcal{L}_{IB}^{(2)}$ can be optimized by minimizing the distance (e.g., KL divergence) between $q(z)$ and $p(z|x, \mathcal{N})$. After obtaining z, we utilize vanilla GCN (Kipf and Welling 2017) with ϕ to parameterize $q(y|z, \mathcal{N})$. Thus, the $\mathcal{L}_{IB}^{(1)}$ can be instantiated as follows for node classification task:

$$\mathcal{L}_{IB}^{(1)} = \mathbb{E}_{p(z,\mathcal{N},y)} \text{Cross-Entropy}(\mathrm{GCN}_\phi(z, \mathcal{N}), y). \tag{9}$$

Federated training with transformed graphs. For each client j, we can obtain a new subgraph $G_j^t = (A_j, Z_j, Y_j)$ after local graph transformation. The Z_j with the best validation performance is stored locally. Then, the client utilizes G_j^t to perform FGC using the general framework in Section 3.1. However, there exist two major flaws: (1) lack of enough labeled samples to obtain high-quality Z_j. Since each client only owns a subgraph of the entire graph, the labeled nodes in each client are scarce, making it hard to apply IB principles (Dai et al. 2023). (2) the obtained Z of different clients are not in the same feature space. To better protect privacy, the graph transformation is conducted locally, i.e., there is no parameter sharing across clients, leading to a heterogeneous feature space and undermining the subsequent FGC. To address the first flaw, we propose a federated self-train strategy to expand labeled node sets. Before local graph transformation, all clients first collaboratively train a GNN model ϕ_{st}. Then, each client utilizes ϕ_{st} to label local unlabeled nodes. To address the second flaw, we propose to initialize the same ϕ among clients with pre-trained ϕ_{st} and fix it during federated training. Each client only needs to locally update $W^{(1)}$ and $W^{(2)}$. In this way, Z in different clients are forced to be unified in the same space.

3.3 Privacy and Utility Analysis

We theoretically analyze the privacy- and utility-preserving properties of FedGC. Let $\mathcal{G}' = (A', X')$, $\mathcal{G}_i = (A_i, X_i)$, $\mathcal{G}_i^t = (A_i, Z_i)$ denote condensed graph, client i's original graph and transformed graph respectively. Assuming $Y_i \to \mathcal{G}_i \to \mathcal{G}_i^t \to \mathcal{G}'$ form a Markov chain, the attack aims to find a graph g to match $\mathcal{G}_i$ with an inference cost $C(\mathcal{G}_i, g)$.

Definition 1. *Inference cost gain* (du Pin Calmon and Fawaz 2012). Inference cost gain measures the quantitative improvement of inferring private data S after observing R, denoted as $\Delta C = c_S^* - \mathbb{E}_{P_R}[c_R^*]$, where c_S^* and c_R^* is the minimum average cost of infer S w/o and w/ R.

In our case, the attack aims to infer $\mathcal{G}_i$ after observing $\mathcal{G}'$. Therefore, we should minimize $\Delta C = c_{\mathcal{G}_i}^* - \mathbb{E}_{P_{\mathcal{G}'}}\left[c_{\mathcal{G}'}^*\right]$. Inspired by (Makhdoumi et al. 2014), we have:

Lemma 1. If $L = \sup|C(\mathcal{G}_i, g)| < \infty$ is satisfied, then $\Delta C = c_{\mathcal{G}_i}^* - \mathbb{E}_{P_{\mathcal{G}'}}\left[c_{\mathcal{G}'}^*\right] \leq 2\sqrt{2}L\sqrt{\mathrm{I}(\mathcal{G}_i; \mathcal{G}')}$.

Lemma 1 demonstrates that ΔC is upperbounded by a function of $\mathrm{I}(\mathcal{G}_i; \mathcal{G}')$ under any bounded C.

Lemma 2. Minimizing $\mathrm{I}(\mathcal{G}_i; \mathcal{G}_i^t)$ is equivalent to minimizing $\mathrm{I}(A_i, X_i; Z_i)$.

Thus, given that $\mathrm{I}(\mathcal{G}_i; \mathcal{G}') \leq \mathrm{I}(\mathcal{G}_i; \mathcal{G}_i^t)$ deduced from the Markov chain, we have the following proposition.

Proposition 1. By minimizing $\mathrm{I}(\mathcal{G}_i; \mathcal{G}_i^t)$, the *inference cost gain* is also minimized.

Therefore, minimizing $\mathrm{I}(A_i, X_i; Z_i)$ could defend the MIA based on the *inference cost gain* theory. As for utility, since $\mathcal{G}'$ stems from $\mathcal{G}_i^t$, we can directly maximum the utility of $\mathcal{G}_i^t$, i.e., maximize $\mathrm{I}(A_i, Z_i; Y_i)$. Noting that this maximization will increase the lower-bound of $\mathrm{I}(A_i, X_i; Z_i)$ in light of $\mathrm{I}(\mathcal{G}_i^t; Y_i) \leq \mathrm{I}(\mathcal{G}_i^t; \mathcal{G}_i)$, thus the objective in Eq.(4) can be seen as a trade-off between privacy- and utility-preserving of the condensed graph, which is controlled by γ. We will present this effect in the experiments.

4 Experiments

4.1 Experimental Setup

Datasets. Follow (Jin et al. 2022b; Zheng et al. 2023), we evaluate FedGC on five graph datasets on node classification task, including Cora, Citeseer, Ogbn-arxiv, Flickr, and Reddit. We adopt the public splits provided in (Jin et al. 2022b).

Baselines. We choose three kinds of baselines to evaluate the effectiveness of condensed graphs by FedGC, including (1) *centralized GCN* (Kipf and Welling 2016), (2) *centralized GC methods*: GCOND (Jin et al. 2022b), DosCond (Jin et al. 2022a) and SFGC (Zheng et al. 2023), and (3) *FGL methods*: BDS-GCN (Wan et al. 2022), FedSage+ (Baek et al. 2023) and FedGCN (Yao et al. 2023). Following (Dai et al. 2023), we replace IB principles with three defense methods to evaluate the effectiveness of FedGC in defending MIA, including PL (Lee et al. 2013), Reg (Nasr, Shokri, and Houmansadr 2018) and LDP (Choi et al. 2018).

Implementation. Following (Jin et al. 2022b), we report results under different condensation ratio r. Following (Yao et al. 2023), we test all FGL methods under the non-i.i.d. depicted by Dirichlet distribution (β=1). we set the client number n=10 for small datasets Cora and Citesser, and n=5 for large-scale datasets Ogbn-arxiv, Flickr, and Reddit. We run 5 times and report the average and variance of results. We utilize accuracy (Acc) to evaluate the condensation performance and AUC score to measure MIA performance.

4.2 Overall Performance

Performance on condensed graph. Table 1 shows the overall performance of the condensed graph compared to baselines. We have the following observations: (1) FedGC achieves superior performance among all baselines, demonstrating the effectiveness of our framework, which also sheds light on new ways to perform GC and FGL tasks. (2) Surprisingly, FedGC performs even better than FGL methods in the non-i.i.d. setting, given that the non-i.i.d. among subgraphs is a key challenge in FGL (Baek et al. 2023). We attribute this to the condensed graph's ability to capture common knowledge across different subgraphs, and the trained model can be further fine-tuned by local data to achieve better personalization. (3) FedGC achieves comparable results with centralized GCN and GC methods, especially in large-scale datasets. Compared to centralized GCN, the gains may lie in that the condensed graph captures general graph patterns that would not be affected by local noisy structures. In terms of GC methods, this superiority is mainly attributed to the fine-tuning strategy, which utilizes personalized model to fit different data distributions.

Performance on MIA. We replace IB with other defense baselines to evaluate the effects of defending MIA. From Table 2, we observe that baselines can only achieve comparable results on the utility of condensed graphs but suffer a dramatic performance decline in defending MIA. By contrast, FedGC can simultaneously achieve high utility and preserve membership privacy. Since the condensed graph is public, model-based defenses (Reg and LDP) can only affect the model gradient matching stage, which has little intervention on the properties entailed by original graphs, leading to

Dataset	Ratio (r)	Graph Condensation (GC)			Federated Graph Learning (FGL)			FedGC	GCN
		GCOND	DOSCOND	SFGC	FedGCN	FedSage+	BDS-GCN		
Cora	1.3%	79.8±1.3	80.0±1.4	80.1±0.4	80.6±0.4	80.3±0.5	76.0±1.4	**81.0±1.1**	81.2±0.2
	2.6%	80.1±0.6	79.2±0.1	**81.7±0.5**				80.8±1.8	
	5.2%	79.3±0.3	79.3±1.3	81.6±0.8				**82.0±0.7**	
Citeseer	0.9%	70.5±1.2	71.4±0.2	71.4±0.5	69.3±0.7	69.8±1.0	67.1±1.8	**71.6±0.1**	71.7±0.1
	1.8%	70.6±0.9	70.5±0.5	**72.4±0.4**				70.5±1.9	
	3.6%	69.8±1.4	70.5±1.1	70.6±0.7				**70.7±1.0**	
Ogbn-arxiv	0.05%	59.2±1.1	59.1±0.8	65.5±0.7	70.8±1.7	70.5±0.5	68.5±0.7	**75.3±0.8**	71.4±0.1
	0.25%	63.2±0.3	60.4±0.9	66.1±0.4				**75.8±0.2**	
	0.5%	64.0±0.4	59.8±0.7	66.8±0.4				**75.3±0.8**	
Flickr	0.1%	49.5±0.1	48.0±0.1	50.1±0.2	51.4±0.2	51.2±0.1	49.9±0.5	**56.2±2.4**	54.3±0.1
	0.5%	49.7±0.1	49.1±0.2	49.5±0.2				**55.1±1.2**	
	1%	49.2±0.1	49.1±0.1	50.5±0.1				**57.1±1.1**	
Reddit	0.05%	89.8±0.1	89.4±0.6	90.3±0.3	91.6±0.4	90.7±0.4	91.4±0.2	**94.2±0.2**	94.5±0.0
	0.1%	89.8±0.2	90.0±0.4	91.2±0.4				**94.3±0.0**	
	0.2%	92.2±0.1	91.6±0.5	92.1±0.1				**94.2±0.2**	

Table 1: Overall performance (%) of condensed graphs by FedGC under different condensation rations compared with baselines.

	Metrics	–	PL	Reg	LDP	FedGC
Cora 2.6%	ACC ↑	79.4	81.2	79.1	78.6	80.8
	AUC ↓	76.4	63.3	72.3	71.4	53.2
Citeseer 1.8%	ACC ↑	71.3	71.7	71.2	71.9	70.5
	AUC ↓	87.2	68.4	83.4	85.1	55.0
Ogbn-arxiv 0.25%	ACC ↑	73.2	73.8	72.5	73.8	75.8
	AUC ↓	64.0	60.2	62.1	63.2	51.6
Flickr 0.5%	ACC ↑	54.0	55.5	52.8	55.2	55.1
	AUC ↓	62.1	59.8	60.7	62.0	54.8
Reddit 0.1%	ACC ↑	93.1	93.4	92.0	93.7	94.3
	AUC ↓	61.1	57.3	60.9	61.2	52.3

Table 2: Overall performance (%) of defending MIA by FedGC compared with baselines. **–** means removing IB.

	Methods	MLP	AP.	Cheby	GCN	SAGE	SGC	Avg.
Cora 2.6%	GCOND	73.1	78.5	76.0	80.1	78.2	79.3	78.4
	SFGC	81.1	78.8	79.0	81.1	81.9	79.1	80.3
	FedGC	77.6	81.2	79.2	80.8	81.9	81.6	81.0
Cite-seer 1.8%	GCOND	63.9	69.6	68.3	70.5	66.2	70.3	69.0
	SFGC	71.3	70.5	71.8	71.6	71.7	71.8	71.5
	FedGC	66.1	71.2	70.5	70.5	71.1	70.2	70.7
Ogbn-arxiv 0.25%	GCOND	62.2	63.4	54.9	63.2	62.6	63.7	61.6
	SFGC	65.1	63.9	60.7	65.1	64.8	64.8	64.3
	FedGC	71.0	75.9	75.7	75.8	76.0	75.0	75.7

Table 3: Generalizability (%) of condensed graphs by FedGC. **AP.**:APPNP. **Avg.**: the average test accuracy across different architectures (excluding MLP).

a high success of MIA. PL instead modifies the original data (adding pseudo labels), thus obtaining better defense. However, the pseudo labels inherit the knowledge of training labels, making MIA still effective. Leveraging IB principles, FedGC transforms original node features into a new space, preventing strong attackers from concluding memberships.

4.3 Generalizability of Condensed Graphs

Table 3 presents the generalizability of condensed graphs, i.e., we use the condensed graph to train different GNN models for testing local data. Following (Jin et al. 2022b), we choose APPNP (Klicpera, Bojchevski, and Günnemann 2019), Cheby (Defferrard, Bresson, and Vandergheynst 2016), GCN, GraphSAGE (Hamilton, Ying, and Leskovec 2017) and SGC (Wu et al. 2019) to report the average performance. We also include MLP for comparison. We can find that FedGC shows good generalizability across different GNN models. It outperforms GCOND in almost all datasets and achieves comparable and even superior performance than the SOTA method SFGC. The root cause is the similar filtering behaviors of these GNN models, which have been illustrated in studies (Jin et al. 2022b; Zhu et al. 2021). The generalizability of condensed graphs enables clients to train personalized models based on self-conditions, which provides a flexible way to conduct downstream tasks.

4.4 Ablation Study

Ablation study on condensed graph. We present the effects of different modules on condensed graphs, shown in Figure 5. *-com*, *-ft*, *-ib*, and *-st* means removing the communications, fine-tuning, local graph transformation, and self-training respectively. We can see that (1) One-step communications contribute more to relatively small graphs (Cora and Citeseer). The reason is that smaller graphs suffer from more information losses (missing significant cross-client neighbors), leading to a cascading effect on the condensed graph (Wang et al. 2022). Similarly, self-training also plays a key role in smaller graphs since they need more labeled nodes to compensate for broken structures. (2) IB can im-

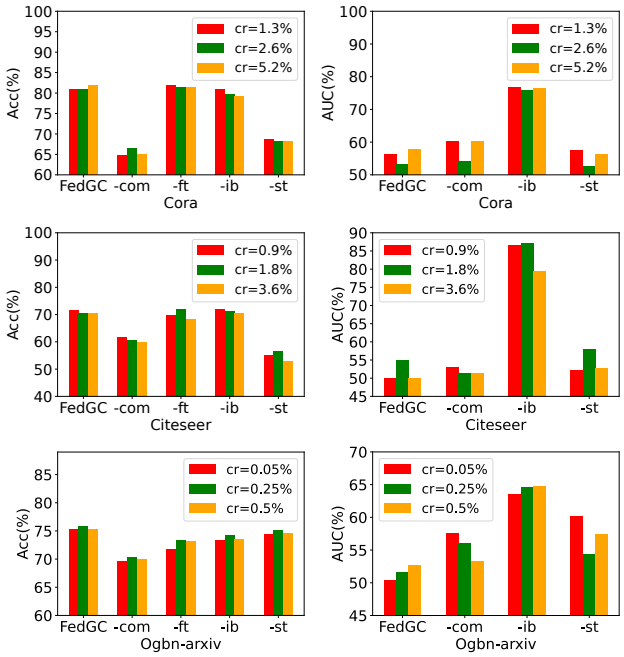

Figure 5: Ablation study for the performance of condensed graph (left ↑) and MIA (right ↓).

prove the generalization ability of the condensed graph, which can be observed that removing IB will suffer from slight performance decay on almost all datasets. We instantiate IB by sampling Z from the original feature distribution (Eq. (8)) and force it to be similar to the standard normal distribution, which avoids over-fitting with local data. (3) Fine-tuning with local data can achieve better personalization. Due to IB principles, the condensed graph captures the general knowledge of multiple subgraphs rather than specific local structure patterns. Thus, a few rounds of fine-tuning by local data can achieve better results.

Ablation study on MIA. The ablation study on MIA is depicted in Figure 5. We remove the *-ft* because it's conducted in the test phase and has no impact on MIA. We can see that the local graph transformation with IB has a significant defensive ability to MIA (up to 40% MIA performance decay on Citeseer). As discussed before, the optimization of the IB objective in Eq. (4) will minimize inference cost gain, making the MIA less effective. It can also be seen that one-step communication and self-training defend MIA to some degree. One-step communication can be seen as a feature augmentation and self-training aims to augment labels. These augmentations make the condensed graph fit to augmented data rather than original data thus alleviating the over-fitting, leading to a performance decay of MIA.

Performance during federated training. We also present the training curve of the condensed graph and MIA w/ and w/o IB, shown in Figure 6. We can see that without IB (*FedGC-IB*) the model converges slowly and also obtains lower performance than the model with IB (*FedGC*). The reason for fast convergence lies in using self-training to label

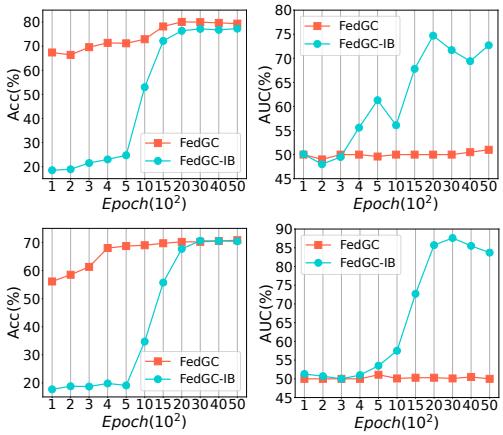

Figure 6: Training curve of condensed graphs (left ↑) and MIA (right ↓) w/o IB. (up: Cora, down: Citeseer).

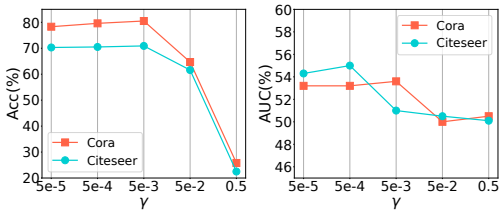

Figure 7: Performance of condensed graphs (left ↑) and MIA (right ↓) under different γ.

nodes before transforming graphs, which introduces more supervised signals to train the IB and consequently incorporates more accurate graph information into Z. The defense of MIA also benefits from the IB. FedGC consistently maintains a lower AUC score during the whole training process.

4.5 Parameter Analysis

Analysis of different γ. Figure 7 presents the performance of condensed graph and MIA under different γ of IB. A proper γ can improve the generalization ability, but a large γ will force Z to follow specific distributions and perform poorly on downstream tasks. The MIA also has a similar trend. A smaller γ causes the model to over-fit the training data, thereby enhancing MIA. On the contrary, a larger γ makes Z less similar to the original features, thus undermining the MIA. Overall, γ controls the trade-off between utility and privacy, which can be flexibly adjusted in reality.

5 Conclusion

In this paper, we investigate the novel problem of federated graph condensation. A general framework by matching weighted aggregated gradients is proposed to preserve the utility of condensed graphs. We also reveal the condensed graph will leak the membership privacy of local data and then propose a local graph transformation with information bottleneck principles to protect privacy. Empirical and theoretical analysis demonstrate the effectiveness of our method.

Acknowledgments

This work is supported in part by the National Key Research and Development Program of China (2023YFF0725103), the National Natural Science Foundation of China (U22B2038, 62192784), the JSPS KAKENHI JP23K24851, the JST PRESTO JPMJPR23P5, and the JST CREST JPMJCR21M2.

References

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

Baek, J.; Jeong, W.; Jin, J.; Yoon, J.; and Hwang, S. J. 2023. Personalized Subgraph Federated Learning. In *ICML*, volume 202 of *Proceedings of Machine Learning Research*, 1396–1415. PMLR.

Brakerski, Z.; Gentry, C.; and Vaikuntanathan, V. 2014. (Leveled) Fully Homomorphic Encryption without Bootstrapping. *ACM Trans. Comput. Theory*, 6(3): 13:1–13:36.

Choi, W.; Tomei, M.; Vicarte, J. R. S.; Hanumolu, P. K.; and Kumar, R. 2018. Guaranteeing Local Differential Privacy on Ultra-Low-Power Systems. In *ISCA*, 561–574. IEEE Computer Society.

Dai, E.; Cui, L.; Wang, Z.; Tang, X.; Wang, Y.; Cheng, M. X.; Yin, B.; and Wang, S. 2023. A Unified Framework of Graph Information Bottleneck for Robustness and Membership Privacy. In *KDD*, 368–379. ACM.

Defferrard, M.; Bresson, X.; and Vandergheynst, P. 2016. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In *NIPS*, 3837–3845.

Dong, T.; Zhao, B.; and Lyu, L. 2022. Privacy for Free: How does Dataset Condensation Help Privacy? In *ICML*, volume 162 of *Proceedings of Machine Learning Research*, 5378–5396. PMLR.

du Pin Calmon, F.; and Fawaz, N. 2012. Privacy against statistical inference. In *Allerton Conference*, 1401–1408. IEEE.

Gao, D.; Yao, X.; and Yang, Q. 2022. A Survey on Heterogeneous Federated Learning. *CoRR*, abs/2210.04505.

Hamilton, W. L.; Ying, Z.; and Leskovec, J. 2017. Inductive Representation Learning on Large Graphs. In *NIPS*, 1024–1034.

Jin, W.; Tang, X.; Jiang, H.; Li, Z.; Zhang, D.; Tang, J.; and Yin, B. 2022a. Condensing Graphs via One-Step Gradient Matching. In *KDD*, 720–730. ACM.

Jin, W.; Zhao, L.; Zhang, S.; Liu, Y.; Tang, J.; and Shah, N. 2022b. Graph Condensation for Graph Neural Networks. In *ICLR*. OpenReview.net.

Kingma, D. P.; and Welling, M. 2014. Auto-Encoding Variational Bayes. In *ICLR*.

Kipf, T. N.; and Welling, M. 2016. Semi-Supervised Classification with Graph Convolutional Networks. *CoRR*, abs/1609.02907.

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

Klicpera, J.; Bojchevski, A.; and Günnemann, S. 2019. Predict then Propagate: Graph Neural Networks meet Personalized PageRank. In *ICLR (Poster)*. OpenReview.net.

Lee, D.-H.; et al. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, 896. Atlanta.

Li, Z.; and Hoiem, D. 2016. Learning Without Forgetting. In *ECCV (4)*, volume 9908 of *Lecture Notes in Computer Science*, 614–629. Springer.

Liu, M.; Li, S.; Chen, X.; and Song, L. 2022. Graph Condensation via Receptive Field Distribution Matching. *CoRR*, abs/2206.13697.

Liu, Y.; Bo, D.; and Shi, C. 2024. Graph Distillation with Eigenbasis Matching. In *ICML*. OpenReview.net.

Makhdoumi, A.; Salamatian, S.; Fawaz, N.; and Médard, M. 2014. From the Information Bottleneck to the Privacy Funnel. In *ITW*, 501–505. IEEE.

Nasr, M.; Shokri, R.; and Houmansadr, A. 2018. Machine Learning with Membership Privacy using Adversarial Regularization. In *CCS*, 634–646. ACM.

Nasr, M.; Shokri, R.; and Houmansadr, A. 2019. Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning. In *IEEE Symposium on Security and Privacy*, 739–753. IEEE.

Olatunji, I. E.; Nejdl, W.; and Khosla, M. 2021. Membership Inference Attack on Graph Neural Networks. In *TPS-ISA*, 11–20. IEEE.

Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *ICLR (Poster)*. OpenReview.net.

Voigt, P.; and Von dem Bussche, A. 2017. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 10(3152676): 10–5555.

Wan, C.; Li, Y.; Li, A.; Kim, N. S.; and Lin, Y. 2022. BNS-GCN: Efficient Full-Graph Training of Graph Convolutional Networks with Partition-Parallelism and Random Boundary Node Sampling. In *MLSys*. mlsys.org.

Wang, B.; Li, A.; Pang, M.; Li, H.; and Chen, Y. 2022. GraphFL: A Federated Learning Framework for Semi-Supervised Node Classification on Graphs. In *ICDM*, 498–507. IEEE.

Wu, F.; Jr., A. H. S.; Zhang, T.; Fifty, C.; Yu, T.; and Weinberger, K. Q. 2019. Simplifying Graph Convolutional Networks. In *ICML*, volume 97 of *Proceedings of Machine Learning Research*, 6861–6871. PMLR.

Wu, T.; Ren, H.; Li, P.; and Leskovec, J. 2020. Graph Information Bottleneck. In *NeurIPS*.

Yang, Z.; Zhang, Y.; Zheng, Y.; Tian, X.; Peng, H.; Liu, T.; and Han, B. 2023. FedFed: Feature Distillation against Data Heterogeneity in Federated Learning. In *NeurIPS*.

Yao, Y.; Jin, W.; Ravi, S.; and Joe-Wong, C. 2023. FedGCN: Convergence-Communication Tradeoffs in Federated Training of Graph Convolutional Networks. In *NeurIPS*.

Zhang, K.; Yang, C.; Li, X.; Sun, L.; and Yiu, S. 2021. Subgraph Federated Learning with Missing Neighbor Generation. In *NeurIPS*, 6671–6682.

Zhang, M.; Pan, S.; Chang, X.; Su, S.; Hu, J.; Haffari, G.; and Yang, B. 2022. BaLeNAS: Differentiable Architecture Search via the Bayesian Learning Rule. In *CVPR*, 11861–11870. IEEE.

Zhao, B.; Mopuri, K. R.; and Bilen, H. 2021. Dataset Condensation with Gradient Matching. In *ICLR*. OpenReview.net.

Zheng, X.; Zhang, M.; Chen, C.; Nguyen, Q. V. H.; Zhu, X.; and Pan, S. 2023. Structure-free Graph Condensation: From Large-scale Graphs to Condensed Graph-free Data. In *NeurIPS*.

Zhu, M.; Wang, X.; Shi, C.; Ji, H.; and Cui, P. 2021. Interpreting and Unifying Graph Neural Networks with An Optimization Framework. In *WWW*, 1215–1226. ACM / IW3C2.