Globally Consistent Federated Graph Autoencoder for Non-IID Graphs

Kun Guo[1] , **Yutong Fang**[1] , **Qingqing Huang**[1] , **Yuting Liang**[1] , **Ziyao Zhang**[1] , **Wenyu He**[1] , **Liu Yang**[2] , **Kai Chen**[2] , **Ximeng Liu**[1] and **Wenzhong Guo**[1]

[1]College of Computer and Data Science, Fuzhou University, Fuzhou 350108, China
[2]Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong 999077, China
gukn@fzu.edu.cn, {fytmolly, hhuangqqingqqing, liangyt2000, zzy_1574, hewenyu2022}@163.com, {lyangau, kaichen}@cse.ust.hk, snbnix@gmail.com, guowenzhong@fzu.edu.cn

Abstract

Graph neural networks (GNNs) have been applied successfully in many machine learning tasks due to their advantages in utilizing neighboring information. Recently, with the global enactment of privacy protection regulations, federated GNNs have gained increasing attention in academia and industry. However, the graphs owned by different participants could be non-independently-and-identically distributed (non-IID), leading to the deterioration of federated GNNs' accuracy. In this paper, we propose a globally consistent federated graph autoencoder (GCFGAE) to overcome the non-IID problem in unsupervised federated graph learning via three innovations. First, by integrating federated learning with split learning, we train a unique global model instead of FedAvg-styled global and local models, yielding results consistent with that of the centralized GAE. Second, we design a collaborative computation mechanism considering overlapping vertices to reduce communication overhead during forward propagation. Third, we develop a layer-wise and block-wise gradient computation strategy to reduce the space and communication complexity during backward propagation. Experiments on real-world datasets demonstrate that GCFGAE achieves not only higher accuracy but also around 500 times lower communication overhead and 1000 times smaller space overhead than existing federated GNN models.

1 Introduction

Graph neural networks (GNNs) have gained broad application in many fields [Abadal *et al.*, 2021; Wu *et al.*, 2021; Wu *et al.*, 2022]. In the real world, large graphs are often spread across multiple participants. For example, a social network is comprised of many personal social circles. Therefore, multiple participants can collaborate to train a global model by aggregating their local graphs, as depicted in Figure 1. It is common that multiple participants' graphs have overlapping vertices, as shown by the grey vertices. Recently, with the increasing concern for personal privacy, it is now legislated in many countries that no private data can be exported without its owner's consent [Jia *et al.*, 2018; Baik, 2020], imposing a significant obstacle to effective distributed graph learning. To tackle the issue, federated learning has emerged as a promising paradigm for developing privacy-preserving distributed graph learning approaches, notably federated GNNs [Liu and Yu, 2022; Fu *et al.*, 2022]. Figure 1 shows a typical scenario of unsupervised federated graph learning where multiple participants' local graphs share overlapping vertices and the attributes of their vertices are identical [Fu *et al.*, 2022]. However, federated GNNs confront the problem that the graphs owned by different participants could be non-independently-and-identically distributed (non-IID), leading to the deterioration of federated GNNs' accuracy [Xie *et al.*, 2021]. In this paper, we focus on a specific non-IID problem in unsupervised federated graph learning, the non-independence between participants' local graphs caused by the missing of partial neighbors of overlapping vertices in participants' local graphs, as illustrated in Figure 1.

At present, there are three types of approaches to overcome the non-IID problem in federated graph learning. The first two types of approaches are personalized federated learning and single-model-based methods [Fu *et al.*, 2022], which rely on the improvements of different FedAvg-based approaches [McMahan *et al.*, 2016]. However, the design of FedAvg's training full global and local models simultaneously means that all these improvements are hard to escape the impact of non-IID data. The third type of approaches explore a novel idea that trains a federated graph model via split learning [Thapa *et al.*, 2022], which replaces the FedAvg-styled training of multiple full models with the training of a unique global model by splitting a neural network into two portions that are independently trained by participants and the coordinator, allowing them to train models consistent with those obtained by centralized GNNs [Shan *et al.*, 2021]. Nevertheless, the third-typed approaches have certain weaknesses. First, they do not consider the relationships between overlapping vertices of different participants, leading to possible accuracy loss, especially when the number of overlapping vertices increases with the involvement of more participants. Second, these approaches transmit the intermediate data related to all vertices between participants during each round of training, incurring high communication and space overhead.

In this paper, we propose a globally consistent federated graph autoencoder (GCFGAE) to solve the non-IID problem

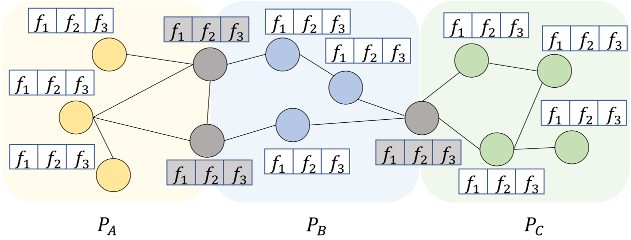

Figure 1: Unsupervised federated graph learning. P_A, P_B, P_C denote three participants, $f_1 \sim f_3$ are the attributes of their local graphs' vertices. Different participants' graphs are rendered with different colors. The grey circles denote overlapping vertices shared between participants.

in unsupervised federated graph learning. First, we construct a distributed GAE framework integrating federated learning and split learning to train a unique global model consistent with that of the centralized GAE. Second, we design a collaborative computation mechanism considering overlapping vertices for the forward propagation of GCFGAE to reduce communication overhead by only conducting the computation related to overlapping vertices between participants and the coordinator. Third, we develop a layer-wise and block-wise gradient computation strategy for the backward propagation of GCFGAE to decompose the computation of gradient matrices into the real-time computation of partial gradient matrices on the coordinator side, the distributed computation of local gradient matrices on the participant side and the combination of them block by block for each neural layer's training. The contributions of this paper are summarized as follows.

- It is the first attempt to overcome the non-IID problem in unsupervised federated graph learning by integrating federated learning with split learning so that the output of GCFGAE is consistent with that of the centralized GAE, that is, GCFGAE does not suffer from accuracy loss.
- The collaborative computation mechanism applied in the forward propagation of GCFGAE achieves significantly lower communication overhead than existing split-learning-based solutions.
- The layer-wise and block-wise gradient computation strategy applied in the backward propagation of GCFGAE can greatly decrease its space overhead.
- We demonstrate the effectiveness of GCFGAE in accuracy preservation and the reduction of communication and space overhead in real-world datasets.

2 Related Work

2.1 Federated Graph Learning on Non-IID Graphs

The non-IID problem in federated graph learning originates from the heterogeneity of the topology and attributes of graphs [Xie *et al.*, 2021]. There are two types of solutions from the perspective of federated learning: personalized federated learning and single-model-based methods [Fu *et al.*, 2022]. Personalized federated learning aims to train personalized models for each participant instead of a global model to circumvent the non-IID problem. GCFL [Xie *et al.*, 2021] and FedCG [Caldarola *et al.*, 2021] clusters participants to improve each participant's local model. FED-PUB [Baek *et al.*, 2022] is a personalized subgraph federated learning framework that jointly improves interrelated local GNN models. FedEgo [Zhang *et al.*, 2022b] applies GraphSAGE [Hamilton *et al.*, 2017] to ego graphs to solve the problem of non-IID labels. The methods based on single models attempt to train more efficient global models by incorporating more information from local models. FedSage [Zhang *et al.*, 2021], FedNI [Peng *et al.*, 2022] and the federated GNN proposed in [Du and Wu, 2022] refine global models by sampling or predicting a vertex's neighbors during each participant's local model training. All the above approaches are developed based on FedAvg and do not focus on the handling of the non-IID problem arises from the missing of partial neighbors of overlapping vertices in participants' local graphs, as shown in Figure 1, leading to possible accuracy loss.

2.2 Federated Learning with Split Learning

Federated learning requires that each participant trains a local model, which is unfriendly to edge devices in the internet of things (IoT) because most edge devices have limited computational resources. Split learning is a feasible complement to federated learning because it distributes the training of different parts of a model among participants and the coordinator. Therefore, integrating federated learning with split learning has received increasing attention [Thapa *et al.*, 2022]. SAPGNN [Shan *et al.*, 2021] is a privacy-preserving GNN developed based on federated learning and split learning that can handle the non-IID problem in supervised federated graph learning. HSFL[Xia *et al.*, 2022] trains a global model based on split federated learning to allocate edge devices' computational resources dynamically. Tian et al. [Tian *et al.*, 2022] proposed a privacy-preserving split learning method for millimeter-wave beam selection. SplitAVG [Zhang *et al.*, 2022a] is a heterogeneity-aware split federated learning method to overcome the performance drops from data heterogeneity in federated learning. RoSFL [Yang *et al.*, 2022] is a robust split federated learning paradigm to avoid model drifting caused by non-IID labels in u-shaped medical image network analysis. Thapa et al. [Thapa *et al.*, 2022] proposed a split federated learning framework to address the issue of limited computational resources in federated learning and training overhead in split learning. All the above approaches require high communication and space overhead because they transmit intermediate data related to all vertices during each round of training. They also do not consider the non-IID problem caused by the overlapping vertices, as shown in Figure 1.

3 Preliminaries

3.1 Centralized GAE

A graph is defined as $G(V, E, \mathbf{A}, \mathbf{X})$, where $V = \{v_1, v_2, ..., v_n\}$ is a set of n vertices, $E = \{(v_i, v_j)\}$ is a set of m links connecting vertices in V. $\mathbf{A} \in \mathbb{R}^{n\times n}$ is the adjacency matrix where $\mathbf{A}_{i,j} = 1$ if there is a link between v_i and v_j. $\mathbf{X} \in \mathbb{R}^{n\times d}$ is the binary attribute matrix where $\mathbf{X}_{i,j} = 1$ if vertex v_i has a value in the jth attribute, and 0 otherwise. Under the scenario of federated graph learning as shown in Figure 1, each participant P_k owns a local graph G_k where a subset of vertices $V_{k,ov}$ are the participant's overlapping vertices shared with others. They collaborate to train a global GNN model by aggregating their local graphs.

We refer to the standard GAE proposed by Kipf and Welling [Kipf and Welling, 2016] as the centralized GAE in this paper and adapt it to federated learning to build GCFGAE. The centralized GAE adopts a GCN-styled encoder which comprises L graph convolutional layers as follows.

$$\mathbf{H}^{(L-1)} = \tilde{\mathbf{A}}\sigma\left(\tilde{\mathbf{A}}\mathbf{X}\mathbf{W}^{(0)}\right)\cdots\mathbf{W}^{(L-1)} \tag{1}$$

where $\tilde{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$, $\tilde{\mathbf{A}}$ is the normalized Laplacian matrix, $\mathbf{D}$ is the degree matrix whose diagonal elements are vertex degrees, $\mathbf{H}^{(L-1)}$ is the $(L-1)$th-layered embedding matrix containing each vertex's embedding vector. σ is the activation function. $\mathbf{W}^{(l)}$ is the weight matrix of the lth layer. The decoder of the centralized GAE is the inner product of the embedding matrix $\mathbf{H}^{(L-1)}$ and itself, as shown in the following equation.

$$\hat{\mathbf{A}} = \sigma\left(\mathbf{H}^{(L-1)}\left(\mathbf{H}^{(L-1)}\right)^{\mathrm{T}}\right) \tag{2}$$

where $\hat{\mathbf{A}}$ is the reconstructed adjacency matrix. The loss function of the centralized GAE is the cross-entropy between the reconstructed and the original adjacency matrices, as shown in the following equation.

$$\mathcal{L} = -\frac{1}{m}\sum_{v_i,v_j}\mathbf{A}_{i,j}\log\hat{\mathbf{A}}_{i,j} + (1-\mathbf{A}_{i,j})\log\left(\hat{\mathbf{A}}_{i,j}\right) \tag{3}$$

where $\hat{\mathbf{A}}_{i,j}$ is the adjacency between v_i and v_j in $\hat{\mathbf{A}}$, respectively.

3.2 Graph Privacy

The attack model studied in this paper is the semi-honest model [Brickell and Shmatikov, 2005] which assumes that each participant follows the routine of a given algorithm and does not collude with the others to capture other participants' privacy. Therefore, as suggested in [Majeed and Lee, 2020], the privacy needed to be protected during federated graph learning are: (1) graph topology including vertex degrees and the existence of links between vertices. Most graphs in the real world, such as social and transportation networks, exhibit the small-world feature, that is, the distribution of vertex degrees is long-tailed, with only a few hub vertices possessing high degrees. An attacker can use the feature to determine the positions of his target vertices; (2) attribute values of vertices and links. The personal profiles in a social network and the points of interest (POIs) of a traveler can be used by an attacker to locate his target vertices, even if he is unaware of a graph's topology.

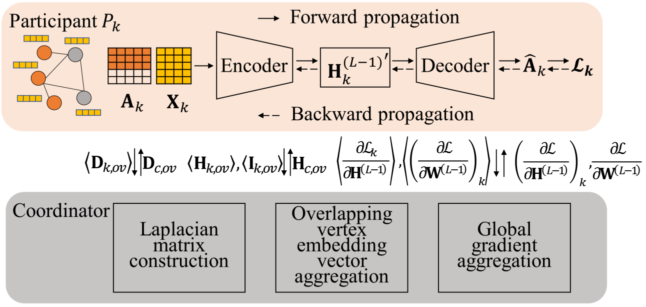
Participant P_k
→ Forward propagation
$\mathbf{A}_k$
$\mathbf{X}_k$
Encoder
$\mathbf{H}_k^{(L-1)'}$
Decoder
$\hat{\mathbf{A}}_k$
$\mathcal{L}_k$
← Backward propagation
$\langle\mathbf{D}_{k,ov}\rangle$
$\mathbf{D}_{c,ov}$
$\langle\mathbf{H}_{k,ov}\rangle, \langle\mathbf{I}_{k,ov}\rangle$
$\mathbf{H}_{c,ov}$
$\left\langle\frac{\partial\mathcal{L}_k}{\partial\mathbf{H}^{(L-1)}}\right\rangle, \left\langle\left(\frac{\partial\mathcal{L}}{\partial\mathbf{W}^{(L-1)}}\right)_k\right\rangle$
$\left(\frac{\partial\mathcal{L}}{\partial\mathbf{H}^{(L-1)}}\right)_k, \frac{\partial\mathcal{L}}{\partial\mathbf{W}^{(L-1)}}$
Coordinator
Laplacian matrix construction
Overlapping vertex embedding vector aggregation
Global gradient aggregation

Figure 2: Framework of GCFGAE

4 The Proposed Model

4.1 Framework of GCFGAE

GCFGAE is developed based on extending the centralized GAE to the unsupervised federated graph learning while executing its forward and backward propagation according to split learning to train a globally consistent model. As shown in Figure 2, GCFGAE is composed of Laplacian matrix construction, the forward propagation relying on the aggregation of overlapping vertices' embedding vectors with the aid of the coordinator and the backward propagation requiring the collaboration of participants and the coordinator in gradient computation. The detailed implementations including mathematical symbols are presented in the following subsections. The analysis of GCFGAE's privacy and consistency and its communication and space complexity is given in the appendix [1] due to the page limit.

4.2 Laplacian Matrix Construction

In order to construct the global Laplacian matrix $\tilde{\mathbf{A}}$ while preserving each participant's privacy, we employ the following steps to compute the degrees of overlapping vertices with the cooperation of participants and the coordinator and to construct a local Laplacian matrix $\tilde{\mathbf{A}}_k$ for each participant. The combination of all $\tilde{\mathbf{A}}_k$s forms the global $\tilde{\mathbf{A}}$.

Step 1: Each participant P_k extracts a matrix $\mathbf{D}_{k,ov}$ corresponding to the degrees of its local overlapping vertices from its local degree matrix $\mathbf{D}_k$ and transforms it into $\langle\mathbf{D}_{k,ov}\rangle$ using secret sharing [Shamir, 1979], where $\langle\cdot\rangle$ denotes the additive secret sharing, and sends it to the coordinator.

Step 2: The coordinator aggregates the $\langle\mathbf{D}_{k,ov}\rangle$s into a global degree matrix $\mathbf{D}_{c,ov} = \sum_k\langle\mathbf{D}_{k,ov}\rangle$ for overlapping vertices and returns it to participants.

Step 3: Participant P_k updates its $\mathbf{D}_{k,ov}$ and $\mathbf{D}_k$ according to $\mathbf{D}_{c,ov}$ by adding up the degrees of the overlapping vertices stored in other participants' graphs and computes its local Laplacian matrix $\tilde{\mathbf{A}}_k = (\mathbf{D}'_k)^{-\frac{1}{2}}\ \mathbf{A}_k\,(\mathbf{D}'_k)^{-\frac{1}{2}}$ where $\mathbf{D}'_k$ is the updated local degree matrix. The combination of all $\tilde{\mathbf{A}}_k$s is $\tilde{\mathbf{A}}$, which is denoted by $\tilde{\mathbf{A}} = ||\tilde{\mathbf{A}}_k$, where $||$ is the combination operator. Specifically, operator $||$ works like building blocks in that it puts the elements of each participant's $\tilde{\mathbf{A}}_k$ in

[1] https://github.com/gcfgae/GCFGAE/blob/main/gcfgae_appendix.pdf

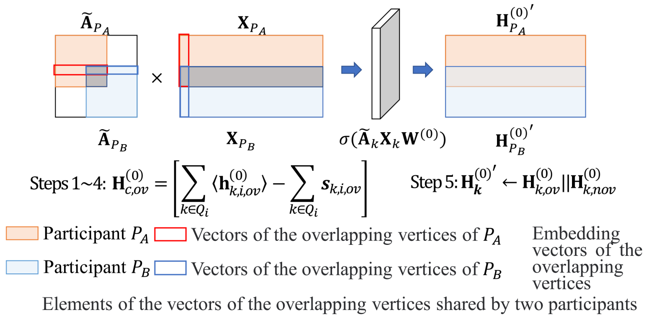
$\widetilde{\mathbf{A}}_{P_A}$
$\mathbf{X}_{P_A}$
$\mathbf{H}_{P_A}^{(0)'}$
×
$\widetilde{\mathbf{A}}_{P_B}$
$\mathbf{X}_{P_B}$
$\sigma(\widetilde{\mathbf{A}}_k\mathbf{X}_k\mathbf{W}^{(0)})$
$\mathbf{H}_{P_B}^{(0)'}$
Steps 1~4: $\mathbf{H}_{c,ov}^{(0)} = \left[\sum_{k\in Q_i}\langle\mathbf{h}_{k,i,ov}^{(0)}\rangle - \sum_{k\in Q_i}\mathbf{s}_{k,i,ov}\right]$
Step 5: $\mathbf{H}_{\mathbf{k}}^{(0)'} \leftarrow \mathbf{H}_{k,ov}^{(0)}||\mathbf{H}_{k,nov}^{(0)}$
Participant P_A
Vectors of the overlapping vertices of P_A
Participant P_B
Vectors of the overlapping vertices of P_B
Embedding vectors of the overlapping vertices
Elements of the vectors of the overlapping vertices shared by two participants

Figure 3: Collaborative computation considering overlapping vertices

their correct positions in $\tilde{\mathbf{A}}$ according to the overlapping and non-overlapping vertices of participants' local graphs to form a complete $\tilde{\mathbf{A}}$.

4.3 Forward Propagation

We design a collaborative computation mechanism considering overlapping vertices to reduce communication overhead during the forward propagation of GCFGAE. Specifically, as illustrated in Figure 3, the collaborative computation for the first layer of GCFGAE consists of two parts: aggregating each participant P_k's local embedding vectors for overlapping vertices $\mathbf{h}_{k,i,ov}^{(0)}$ into a global embedding matrix $\mathbf{H}_{c,ov}^{(0)}$ and constructing a local embedding matrix $\mathbf{H}_k^{(0)'}$ by combining the local embedding matrices $\mathbf{H}_{k,ov}^{(0)}$ and $\mathbf{H}_{k,nov}^{(0)}$ corresponding to overlapping and non-overlapping vertices, respectively. The communication overhead is reduced by only transmitting the intermediate data related to overlapping vertices rather than all vertices. The details are given as follows.

Step 1: All participants share initial weight matrices $\mathbf{W}^{(0)} \sim \mathbf{W}^{(L-1)}$. Notably, only one set of global weight matrices or one global model is updated during the training of GCFGAE since we implement the forward and backward propagation based on split learning.

Step 2: Each participant P_k constructs its local first-layered emedding matrix $\mathbf{H}_k^{(0)} = \sigma\left(\widetilde{\mathbf{A}}_k\mathbf{X}_k\mathbf{W}^{(0)}\right)$.

Step 3: Each participant P_k selects the embedding vector $\mathbf{h}_{k,i,ov}^{(0)}$ of each local overlapping vertex $v_{k,i,ov} \in V_{k,ov}$ from $\mathbf{H}_k^{(0)}$ and transforms it into a secret $\langle\mathbf{h}_{k,i,ov}^{(0)}\rangle$ according to equation 4, where $\mathbf{r}_{k,j}$ is the random vector generated for each participant P_j sharing $v_{k,i,ov}$ with P_k. The indices of all such participants are put in an index set Q_i. A secret local embedding matrix $\langle\mathbf{H}_{k,ov}^{(0)}\rangle$ is constructed by packing all $\langle\mathbf{h}_{k,i,ov}^{(0)}\rangle$s into a marix. A subtle problem here is that equation 4 counts the embedding vector of an overlapping vertex $v_{k,i,ov}$ one more time for each participant $P_j (j \in Q_i)$ sharing the vertex with P_k. Therefore, we must record the duplicated terms in the computation of these embedding vectors in a set $S_{k,ov} = \{\mathbf{s}_{k,i,ov}\}$ for the subsequent deduction, where $\mathbf{s}_{k,i,ov}$ is a vector formed by the duplicated terms in the computation of the embedding vectors. Secrets $\langle\mathbf{H}_{k,ov}^{(0)}\rangle$ and $\langle S_{k,ov}\rangle$ are sent to the coordinator.

$$\langle\mathbf{H}_{k,ov}^{(0)}\rangle = \left[\langle\mathbf{h}_{k,i,ov}^{(0)}\rangle\right] = \left[\mathbf{h}_{k,i,ov}^{(0)} - \sum_{j\in Q_i}\mathbf{r}_{k,j}\right] \quad (4)$$

Step 4: The coordinator extracts $\left\langle\mathbf{h}_{k,i,ov}^{(0)}\right\rangle$s corresponding to participant $P_k (k \in Q_i)$ from $\langle\mathbf{H}_{k,ov}^{(0)}\rangle$ for each overlapping vertex $v_{i,ov}$ and aggregates them into a global embedding vector $\mathbf{h}_{c,i,ov}^{(0)}$ according to equation 5. The subtraction operation is used to deduct the duplicated terms from the sum of $\langle\mathbf{h}_{k,i,ov}^{(0)}\rangle$s. All $\mathbf{h}_{c,i,ov}^{(0)}$s are encapsulated into a global embedding matrix $\mathbf{H}_{c,ov}^{(0)}$ that is sent back to each participant.

$$\mathbf{H}_{c,ov}^{(0)} = \left[\mathbf{h}_{c,i,ov}^{(0)}\right] = \left[\sum_{k\in Q_i}\left\langle\mathbf{h}_{k,i,ov}^{(0)}\right\rangle - \sum_{k\in Q_i}\mathbf{s}_{k,i,ov}\right] \quad (5)$$

Step 5: Participant P_k updates its local embedding matrix $\mathbf{H}_{k,ov}^{(0)}$ of overlapping vertices according to $\mathbf{H}_{c,ov}^{(0)}$ and combines it with the local embedding matrix $\mathbf{H}_{k,nov}^{(0)}$ of non-overlapping vertices to construct a complete local embedding matrix $\mathbf{H}_k^{(0)'}$ according to equation 6. $\mathbf{H}_{k,ov}^{(0)}$ and $\mathbf{H}_{k,nov}^{(0)}$ are combined in the same way as that in the construction of $\tilde{\mathbf{A}}$ in subsection 4.2.

$$\mathbf{H}_k^{(0)'} \leftarrow \mathbf{H}_{k,ov}^{(0)}||\mathbf{H}_{k,nov}^{(0)} \quad (6)$$

Steps 2 to 5 are repeated for layers 1 to $(L-1)$ to produce the $(L-1)$th-layered embedding matrix $\mathbf{H}_k^{(L-1)'}$ for each participant.

Step 6: Each participant P_k constructs a local reconstructed adjacency matrix $\hat{\mathbf{A}}_k$ based on $\mathbf{H}_k^{(L-1)'}$ according to equation 2 and computes the local loss $\mathcal{L}_k$ according to equation 3. Then, it sends $\mathbf{H}_k^{(L-1)'}$ to the coordinator. In GCFGAE, the global loss $\mathcal{L}$ is decomposed into each participant's local loss $\mathcal{L}_k$. The gradients of the global loss $\mathcal{L}$ with respect to embedding and weight matrices are computed through the gradients of $\mathcal{L}_k$ with respect to them. The details are presented in the next subsection.

Step 7: The coordinator combines all $\mathbf{H}_k^{(L-1)'}$s to construct a global embedding matrix $\mathbf{H}^{(L-1)}$ with the $||$ operator and computes a global recontructed adjacency matrix $\hat{\mathbf{A}}$ according to equation 2.

4.4 Backward Propagation

We develop a layer-wise and block-wise gradient computation strategy to reduce space overhead during the backward propagation of GCFGAE. Figure 4 illustrates the application of the strategy applies to the backward propagation of the $(L-1)$th and $(L-2)$th layers. Other layers are processed in the same manner as the $(L-2)$th layer. The layer-wise and block-wise gradient computation at the $(L-1)$th layer

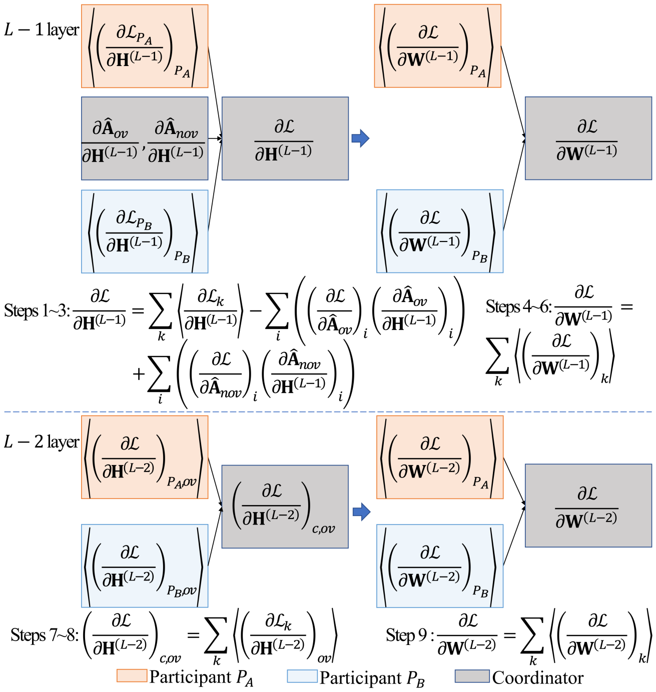
$L-1$ layer
$\left\langle \left(\frac{\partial \mathcal{L}_{P_A}}{\partial \mathbf{H}^{(L-1)}} \right) \right\rangle_{P_A}$
$\frac{\partial \hat{\mathbf{A}}_{ov}}{\partial \mathbf{H}^{(L-1)}}, \frac{\partial \hat{\mathbf{A}}_{nov}}{\partial \mathbf{H}^{(L-1)}}$
$\left\langle \left(\frac{\partial \mathcal{L}_{P_B}}{\partial \mathbf{H}^{(L-1)}} \right) \right\rangle_{P_B}$
$\frac{\partial \mathcal{L}}{\partial \mathbf{H}^{(L-1)}}$
$\left\langle \left(\frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(L-1)}} \right) \right\rangle_{P_A}$
$\left\langle \left(\frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(L-1)}} \right) \right\rangle_{P_B}$
$\frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(L-1)}}$
Steps 1~3: $\frac{\partial \mathcal{L}}{\partial \mathbf{H}^{(L-1)}} = \sum_k \left\langle \frac{\partial \mathcal{L}_k}{\partial \mathbf{H}^{(L-1)}} \right\rangle - \sum_i \left(\left(\frac{\partial \mathcal{L}}{\partial \hat{\mathbf{A}}_{ov}} \right)_i \left(\frac{\partial \hat{\mathbf{A}}_{ov}}{\partial \mathbf{H}^{(L-1)}} \right)_i \right) + \sum_i \left(\left(\frac{\partial \mathcal{L}}{\partial \hat{\mathbf{A}}_{nov}} \right)_i \left(\frac{\partial \hat{\mathbf{A}}_{nov}}{\partial \mathbf{H}^{(L-1)}} \right)_i \right)$
Steps 4~6: $\frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(L-1)}} = \sum_k \left\langle \left(\frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(L-1)}} \right)_k \right\rangle$
$L-2$ layer
$\left\langle \left(\frac{\partial \mathcal{L}}{\partial \mathbf{H}^{(L-2)}} \right)_{P_A ov} \right\rangle$
$\left\langle \left(\frac{\partial \mathcal{L}}{\partial \mathbf{H}^{(L-2)}} \right)_{P_B ov} \right\rangle$
$\left(\frac{\partial \mathcal{L}}{\partial \mathbf{H}^{(L-2)}} \right)_{c,ov}$
$\left\langle \left(\frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(L-2)}} \right) \right\rangle_{P_A}$
$\left\langle \left(\frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(L-2)}} \right) \right\rangle_{P_B}$
$\frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(L-2)}}$
Steps 7~8: $\left(\frac{\partial \mathcal{L}}{\partial \mathbf{H}^{(L-2)}} \right)_{c,ov} = \sum_k \left\langle \left(\frac{\partial \mathcal{L}_k}{\partial \mathbf{H}^{(L-2)}} \right)_{ov} \right\rangle$
Step 9: $\frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(L-2)}} = \sum_k \left\langle \left(\frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(L-2)}} \right)_k \right\rangle$
🟧 Participant P_A
🟦 Participant P_B
⬜ Coordinator

Figure 4: Layer-wise and block-wise gradient computation

consists of two parts: the computation of the gradient matrix of the global loss $\mathcal{L}$ with respect to the $(L-1)$th-layered embedding matrix $\mathbf{H}^{(L-1)}$ and the computation of the gradient matrix of $\mathcal{L}$ with respect to the $(L-1)$th-layered weight matrix $\left(\mathbf{W}^{(L-1)}\right)$. Similarly, the layer-wise and block-wise gradient computation at the $(L-2)$th layer is also composed of two parts: the computation of the blocks of the gradient matrix of the global loss $\mathcal{L}$ with respect to the $(L-2)$th-layered embedding matrix $\mathbf{H}^{(L-1)}$ corresponding to overlapping vertices and the computation of the gradient matrix of $\mathcal{L}$ with respect to the $(L-2)$th-layered weight matrix $\left(\mathbf{W}^{(L-2)}\right)$. The blocks are computed in a real-time manner to reduce space overhead. The backward propagation of GCFGAE is composed of the following steps.

Step 1: The coordinator computes a set of gradient matrices $(\frac{\partial \hat{\mathbf{A}}}{\partial \mathbf{H}^{(L-1)}})_k$ [2] for each participant P_k based on the reconstructed adjacency matrix $\hat{\mathbf{A}}$ and the $(L-1)$th-layered global embedding matrix $\mathbf{H}^{(L-1)}$ and sends them to participants. Specifically, each gradient matrix in the form of $(\frac{\partial \hat{\mathbf{A}}_{i,j}}{\partial \mathbf{H}^{(L-1)}})_k$, where $\hat{\mathbf{A}}_{i,j}$ is an element of $\hat{\mathbf{A}}$, is transmitted once it has been computed. More importantly, to further reduce the communication overhead, the coordinator only transmits the ith and the jth row vectors of $(\frac{\partial \hat{\mathbf{A}}_{i,j}}{\partial \mathbf{H}^{(L-1)}})_k$ to participants.

Step 2: Each participant P_k computes a local gradient matrix $\frac{\partial \mathcal{L}_k}{\partial \mathbf{H}^{(L-1)}}$ of local loss $\mathcal{L}_k$ with respect to the $(L-1)$th-layered global embedding matrix $\mathbf{H}^{(L-1)}$ according to equation 7. Index k indicates only selecting the gradient matrices related to participant P_k from the gradient matrix set of $\frac{\partial \hat{\mathbf{A}}}{\partial \mathbf{H}^{(L-1)}}$. Index i traverses all selected matrices. Then, participant P_k transforms $\frac{\partial \mathcal{L}_k}{\partial \mathbf{H}^{(L-1)}}$ into a secret $\langle \frac{\partial \mathcal{L}_k}{\partial \mathbf{H}^{(L-1)}} \rangle$ and sends it to the coordinator.

$$\frac{\partial \mathcal{L}_k}{\partial \mathbf{H}^{(L-1)}} = \sum_i \left(\left(\frac{\partial \mathcal{L}_k}{\partial \hat{\mathbf{A}}_k} \right)_i \left(\left(\frac{\partial \hat{\mathbf{A}}}{\partial \mathbf{H}^{(L-1)}} \right)_k \right)_i \right) \tag{7}$$

Step 3: The coordinator first aggregates the received $\langle \frac{\partial \mathcal{L}_k}{\partial \mathbf{H}^{(L-1)}} \rangle$s into $\sum_k \langle \frac{\partial \mathcal{L}_k}{\partial \mathbf{H}^{(L-1)}} \rangle$. Second, it computes the gradient matrix $\frac{\partial \mathcal{L}}{\partial \hat{\mathbf{A}}_{ov}}$ of the global loss $\mathcal{L}$ with respect to the reconstructed adjacency matrix $\hat{\mathbf{A}}_{ov}$ corresponding to overlapping vertices. Third, it computes the gradient matrix $\frac{\partial \mathcal{L}}{\partial \hat{\mathbf{A}}_{nov}}$ of $\mathcal{L}$ with respect to the reconstructed adjacency matrix $\hat{\mathbf{A}}_{nov}$ corresponding to non-overlapping vertices. Forth, it computes the gradient matrix $\frac{\partial \mathcal{L}}{\partial \mathbf{H}^{(L-1)}}$ of $\mathcal{L}$ with respect to $\mathbf{H}^{(L-1)}$ according to equation 8. Index i traverses the gradient matrices of $\frac{\partial \hat{\mathbf{A}}_{ov}}{\partial \mathbf{H}^{(L-1)}}$ and $\frac{\partial \hat{\mathbf{A}}_{nov}}{\partial \mathbf{H}^{(L-1)}}$. The subtraction operation is used to deduct the duplicated terms in $\sum_k \langle \frac{\partial \mathcal{L}_k}{\partial \mathbf{H}^{(L-1)}} \rangle$. The addition operation is used to append the gradients of $\mathcal{L}$ with respect to the embedding vectors of non-overlapping vertices. Finally, the coordinator extracts $\left(\frac{\partial \mathcal{L}}{\partial \mathbf{H}^{(L-1)}}\right)_k$ corresponding to participant P_k from $\frac{\partial \mathcal{L}}{\partial \mathbf{H}^{(L-1)}}$ and sends it to the participant.

$$\frac{\partial \mathcal{L}}{\partial \mathbf{H}^{(L-1)}} = \sum_k \left\langle \frac{\partial \mathcal{L}_k}{\partial \mathbf{H}^{(L-1)}} \right\rangle - \sum_i \left(\left(\frac{\partial \mathcal{L}}{\partial \hat{\mathbf{A}}_{ov}} \right)_i \left(\frac{\partial \hat{\mathbf{A}}_{ov}}{\partial \mathbf{H}^{(L-1)}} \right)_i \right) + \sum_i \left(\left(\frac{\partial \mathcal{L}}{\partial \hat{\mathbf{A}}_{nov}} \right)_i \left(\frac{\partial \hat{\mathbf{A}}_{nov}}{\partial \mathbf{H}^{(L-1)}} \right)_i \right) \tag{8}$$

Step 4: Each pariticpant P_k computes its local gradient matrix $\left(\frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(L-1)}}\right)_k$ according to equation 9 and transforms it into a secret $\langle \left(\frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(L-1)}}\right)_k \rangle$ and sends it to the coordinator.

$$\left(\frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(L-1)}} \right)_k = \left(\frac{\partial \mathcal{L}}{\partial \mathbf{H}^{(L-1)}} \right)_k \frac{\partial \mathbf{H}_k^{(L-1)}}{\partial \mathbf{W}^{(L-1)}} \tag{9}$$

Step 5: The coordinator aggregates all $\langle \left(\frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(L-1)}}\right)_k \rangle$s into $\left(\frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(L-1)}}\right)$ according to equation 10 and sends it to all participants.

$$\left(\frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(L-1)}} \right) = \sum_k \left\langle \left(\frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(L-1)}} \right)_k \right\rangle \tag{10}$$

Step 6: Each participant P_k updates $\mathbf{W}^{(L-1)}$ based on $\left(\frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(L-1)}}\right)$ via stochastic gradient descent (SGD) [Shi *et al.*, 2020].

Step 7: Each participant P_k computes the gradient matrix $\frac{\partial \mathcal{L}_k}{\partial \mathbf{H}^{(L-2)}}$ of local loss $\mathcal{L}_k$ with respect to the $(L-2)$th-layered embedding matrix $\mathbf{H}^{(L-2)}$ according to equation 11 where indices k and i serve the same purpose as in equation 7, and

[2] The gradient of a matrix $\mathbf{A}$ with respect to another matrix $\mathbf{B}$ is a set of n_A gradient matrices each of which is $\frac{\partial \mathbf{A}_{i,j}}{\partial \mathbf{B}}$, where n_A is the number of elements of $\mathbf{A}$, and $\mathbf{A}_{i,j}$ is an element of $\mathbf{A}$.

transforms the blocks in $\frac{\partial \mathcal{L}_k}{\partial \mathbf{H}^{(L-2)}}$ corresponding to overlapping vertices into a secret $\langle (\frac{\partial \mathcal{L}_k}{\partial \mathbf{H}^{(L-2)}})_{ov} \rangle$ and sends it to the coordinator.

$$\frac{\partial \mathcal{L}_k}{\partial \mathbf{H}^{(L-2)}} = \sum_i \left(\left(\left(\frac{\partial \mathcal{L}}{\partial \mathbf{H}^{(L-1)}} \right)_k \right)_i \left(\frac{\partial \mathbf{H}_k^{(L-1)}}{\partial \mathbf{H}_k^{(L-2)}} \right)_i \right) \tag{11}$$

Step 8: The coordinator aggregates all $\langle (\frac{\partial \mathcal{L}_k}{\partial \mathbf{H}^{(L-2)}})_{ov} \rangle$s into $\left(\frac{\partial \mathcal{L}}{\partial \mathbf{H}^{(L-2)}} \right)_{c,ov}$ according to equation 12 and sends it to all participants.

$$\left(\frac{\partial \mathcal{L}}{\partial \mathbf{H}^{(L-2)}} \right)_{c,ov} = \sum_k \left\langle \left(\frac{\partial \mathcal{L}_k}{\partial \mathbf{H}^{(L-2)}} \right)_{ov} \right\rangle \tag{12}$$

Step 9: Participant P_k computes $\left(\frac{\partial \mathcal{L}}{\partial \mathbf{H}^{(L-2)}} \right)_k$ by replacing $(\frac{\partial \mathcal{L}_k}{\partial \mathbf{H}^{(L-2)}})_{ov}$ with $\left(\frac{\partial \mathcal{L}}{\partial \mathbf{H}^{(L-2)}} \right)_{c,ov}$ and merges it with the gradients of $\mathcal{L}$ with respect to the embedding vectors of its local non-overlapping vertices and updates the $(L-2)$th-layered weight matrix $\mathbf{W}^{(L-2)}$ according to steps 4 to 6.

Steps 7 to 9 are repeated for layers $L-3$ to 0 to update the weight matrices corresponding to those layers.

5 Experiments

We have conducted comprehensive experiments on privacy-preserving community detection, a typical unsupervised graph learning task, to evaluate the performance of GFGAE and baseline models. First, we introduce the datasets and evaluation metrics used in the experiments. Second, we present the experimental results on the consistency with the centralized GAE introduced in subsection 3.1, the accuracy and the communication and space overhead compared with baselines. Kmeans [Sooksatra *et al.*, 2022] is used to cluster the embedding vectors output by GCFGAE and baselines. Communities are built by partitioning vertices whose embedding vectors are in the same cluster into the same group.

5.1 Datasets and Evaluation Metrics

Five real-world networks are used in the experiments: (1) Cora: n=2708, m=5278, $avgd$=3.90, w=1433; (2) Citeseer: n=3264, m=4536, $avgd$=2.78, w=3703; (3) DBLP: n=1906, m=6644, $avgd$=6.97, w=45; (4) Amazon: n=2187, m=6413, $avgd$=5.86, w=480; (5) Lastfm: n=1215, m=5707, $avgd$=9.39, w=180, where n, m, $avgd$ and w denote the number of vertices and links, the average degree and the dimension of attribute vectors of a network, respectively. We source Cora and Citeseer from LINQS [3] and DBLP, Amazon and Lastfm_asia (abbreviated as Lastfm) from SNAP [4]. The method in [Gonzalez *et al.*, 2012] is adopted to split a network into 2 to 10 overlapping subnetworks to simulate participants' graphs. As the number of participants increases, the number of overlapping vertices between participants increases, leading to more non-IID graphs. Therefore, varying numbers of participants are used to simulate different degrees of non-IID graphs in the experiments.

Two evaluation metrics, normalized mutual information (NMI) and adjusted rand index (ARI) [Chakraborty *et al.*, 2017], are used to measure models' accuracy. A higher value of NMI of ARI indicates higher accuracy. The communication overhead (CO) and the space overhead (SO) are measured by CO $= n_r \sum_i (n_{M,i} n_b)$ and SO $= \sum_j (n_{M,j} n_b)$, respectively, where n_r is the number of iterations, $n_{M,i}$ is the number of elements in a matrix M_i transmitting between participants and the coordinator, n_b is the bytes of an element, $n_{M,j}$ is the number of elements in a matrix M_j that should be simultaneously stored during an iteration.

5.2 Baselines

We first compared GCFGAE with the centralized GAE introduced in subsection 3.1 to verify that it could achieve consistent results with the centralized GAE. Second, we compared GCFGAE with SAPGNN [Shan *et al.*, 2021], FedAvg+GAE and a simplified distributed GAE (abbreviated as SDGAE) to evaluate its accuracy on graphs with different non-IID degrees and the effectiveness of the integration of federated learning and split learning. SAPGNN is the state-of-the-art GNN framework based on federated and split learning. However, it is designed for supervised federated graph learning. We adapted it to our experiments by replacing its loss computation with equations 2 and 3 and using a GCN-styled encoder similar to GCFGAE. FedAvg+GAE uses the centralized GAE to train a participant's local model and FedAvg to aggregate local models to update the global model in a similar manner to FedSage [Zhang *et al.*, 2021]. SDGAE runs the centralized GAE on each participant's side with no federated learning or split learning mechanism to find local communities and joins them to produce final global communities. The source code of all models is written in Python [5].

5.3 Consistency Experiment

Figure 5 shows the results of the consistency experiment. The number after a model's name indicates the number of participants the model was trained on. It is evident that the accuracy of GCFGAE is identical to that of the centralized GAE regardless of whether it is measured by NMI or ARI, or measured under any number of participants. Therefore, the experimental results clearly suggest that GCFGAE bears no accuracy loss when applying GAE to distributed privacy-preserving community detection based on federated and split learning, confirming our consistency analysis in the appendix. Particularly, the proper computation considering overlapping vertices based on the collaborative computation mechanism and the layer-wise and block-wise gradient computation strategy during the forward and backward propagation of GCFGAE plays a vital role.

5.4 Accuracy Experiment

The results of the accuracy experiment are shown in Figure 6. We report the accuracy measured by NMI because Figure 5 reveals that the accuracy measured by ARI is in accordance with that measured by NMI. As shown in Figure 6 (a) and (b), GCFGAE surpasses all baselines on all datasets, which is

[3] https://linqs.soe.ucsc.edu/data

[4] http://snap.stanford.edu/data

[5] https://github.com/gcfgae/GCFGAE

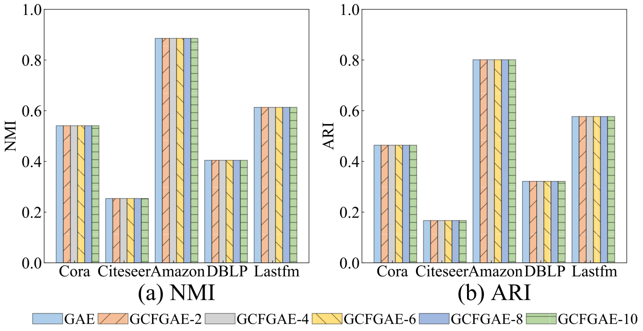

Figure 5: Results of the consistency experiment

largely due to the integration of federated learning with split learning and the specific design of its forward and backward propagation. SAPGNN is inferior to FedAvg+GAE because it ignores the relationships between the overlapping vertices, although we adapt it to unsupervised federated graph learning by replacing its loss function. The accuracy of SDGAE is the lowest in nearly all datasets, demonstrating the significant superiority of federated and split learning over the simple combination of separately running models. Figure 6 (c) and (d) reveal that the non-IID degree of participants' local graphs has no effect on GCFGAE but a substantial impact on baselines, demonstrating the effectiveness of GCFGAE's globally consistent model training.

5.5 Communication and Space Overhead Experiment

We compared GCFGAE with SAPGNN in the communication and space overhead experiments since both utilize federated learning and split learning to train a unique global model, unlike the FedAvg-based models. The results are shown in Figure 7. The logarithm of CO and SO values are used as the vertical axes because of the great difference between the results of GCFGAE and SAPGNN. As shown in Figure 7, the communication overhead of GCFGAE is around 500 times lower than that of SAPGNN, and the space overhead of GCFGAE is nearly 1000 times smaller than that of SAPGNN. The remarkable improvements strongly demonstrate the effectiveness of the collaborative computation mechanism and the layer-wise and block-wise gradient computation strategy used in the forward and backward propagation of GCFGAE.

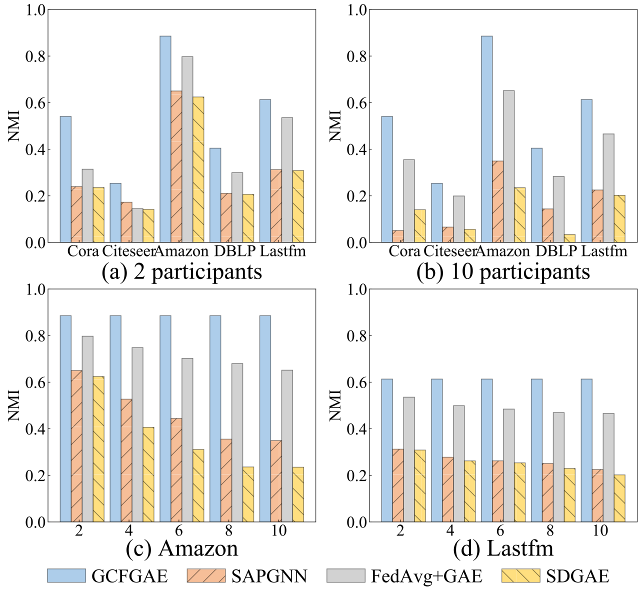

Figure 6: Results of the accuracy experiment

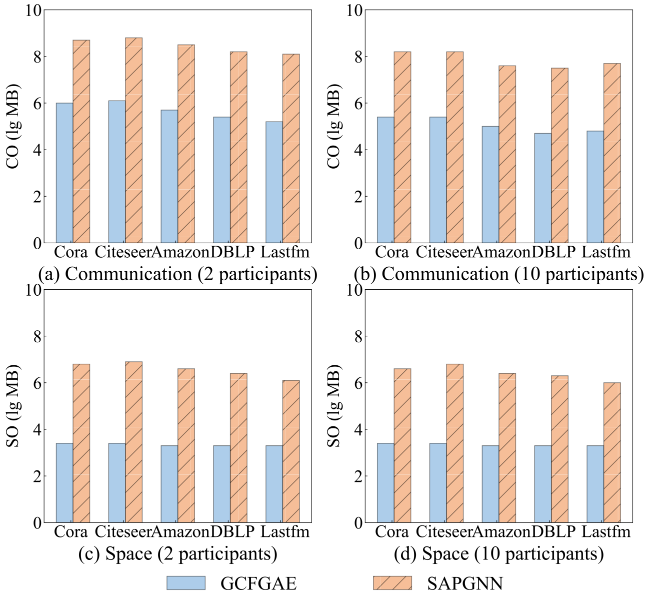

Figure 7: Results of the communication and space overhead experiment

6 Conclusion

In this paper, we propose a federated graph autoencoder GCFGAE to solve the non-IID problem in unsupervised federated graph learning. We achieve lossless accuracy by training a globally consistent model based on the integration of federated learning and split learning. The collaborative computation mechanism considering overlapping vertices and the layer-wise and block-wise gradient computation strategy employed in the forward and backward propagation boost GCFGAE's accuracy and greatly reduce its communication and space overhead. Experiments on real-world datasets demonstrate the effectiveness of GCFGAE.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant No. 62002063 and No. U21A20472, the National Key Research and Development Plan of China under Grant No.2021YFB3600503, the Fujian Collaborative Innovation Center for Big Data Applications in Governments, the Fujian Industry-Academy Cooperation Project under Grant No. 2017H6008 and No. 2018H6010, the Natural Science Foundation of Fujian Province under

Grant No.2022J01118, No.2020J05112 and No.2020J01420, the Fujian Provincial Department of Education under Grant No.JAT190026, the Major Science and Technology Project of Fujian Province under Grant No.2021HZ022007, the Hong Kong RGC TRS T41-603/20R and Haixi Government Big Data Application Cooperative Innovation Center.

References

[Abadal *et al.*, 2021] Sergi Abadal, Akshay Jain, Robert Guirado, Jorge López-Alonso, and Eduard Alarcón. Computing graph neural networks: A survey from algorithms to accelerators. *ACM Computing Surveys (CSUR)*, 54(9):1–38, 2021.

[Baek *et al.*, 2022] Jinheon Baek, Wonyong Jeong, Jiongdao Jin, Jaehong Yoon, and Sung Ju Hwang. Personalized subgraph federated learning. *arXiv preprint arXiv:2206.10206*, 2022.

[Baik, 2020] Jeeyun Sophia Baik. Data privacy against innovation or against discrimination?: The case of the california consumer privacy act (ccpa). *Telematics and Informatics*, 52, 2020.

[Brickell and Shmatikov, 2005] Justin Brickell and Vitaly Shmatikov. Privacy-preserving graph algorithms in the semi-honest model. In *Advances in Cryptology - ASIACRYPT 2005, 11th International Conference on the Theory and Application of Cryptology and Information Security, Chennai, India, December 4-8, 2005, Proceedings*, 2005.

[Caldarola *et al.*, 2021] Debora Caldarola, Massimiliano Mancini, Fabio Galasso, Marco Ciccone, Emanuele Rodolà, and Barbara Caputo. Cluster-driven graph federated learning over multiple domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2749–2758, 2021.

[Chakraborty *et al.*, 2017] Tanmoy Chakraborty, Ayushi Dalmia, Animesh Mukherjee, and Niloy Ganguly. Metrics for community analysis: A survey. *ACM Computing Surveys (CSUR)*, 50(4):1–37, 2017.

[Du and Wu, 2022] Bingqian Du and Chuan Wu. Federated graph learning with periodic neighbour sampling. In *2022 IEEE/ACM 30th International Symposium on Quality of Service (IWQoS)*, pages 1–10. IEEE, 2022.

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

[Gonzalez *et al.*, 2012] Joseph E Gonzalez, Yucheng Low, Haijie Gu, Danny Bickson, and Carlos Guestrin. {PowerGraph}: Distributed {Graph-Parallel} computation on natural graphs. In *10th USENIX symposium on operating systems design and implementation (OSDI 12)*, pages 17–30, 2012.

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

[Jia *et al.*, 2018] Jian Jia, Ginger Zhe Jin, and Liad Wagman. The short-run effects of gdpr on technology venture investment. Technical report, National Bureau of Economic Research, 2018.

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

[Liu and Yu, 2022] Rui Liu and Han Yu. Federated graph neural networks: Overview, techniques and challenges. *arXiv preprint arXiv:2202.07256*, 2022.

[Majeed and Lee, 2020] Abdul Majeed and Sungchang Lee. Anonymization techniques for privacy preserving data publishing: A comprehensive survey. *IEEE access*, 9:8512–8545, 2020.

[McMahan *et al.*, 2016] H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*, 2016.

[Peng *et al.*, 2022] Liang Peng, Nan Wang, Nicha Dvornek, Xiaofeng Zhu, and Xiaoxiao Li. Fedni: Federated graph learning with network inpainting for population-based disease prediction. *IEEE Transactions on Medical Imaging*, 2022.

[Shamir, 1979] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, nov 1979.

[Shan *et al.*, 2021] Chuanqiang Shan, Huiyun Jiao, and Jie Fu. Towards representation identical privacy-preserving graph neural network via split learning. *arXiv preprint arXiv:2107.05917*, 2021.

[Shi *et al.*, 2020] Han Shi, Haozheng Fan, and James T Kwok. Effective decoding in graph auto-encoder using triadic closure. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 906–913, 2020.

[Sooksatra *et al.*, 2022] Korn Sooksatra, Rokin Maharjan, and Tomas Cerny. Monolith to microservices: Vae-based gnn approach with duplication consideration. In *2022 IEEE International Conference on Service-Oriented System Engineering (SOSE)*, pages 1–10. IEEE, 2022.

[Thapa *et al.*, 2022] Chandra Thapa, Pathum Chamikara Mahawaga Arachchige, Seyit Camtepe, and Lichao Sun. Splitfed: When federated learning meets split learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8485–8493, 2022.

[Tian *et al.*, 2022] Muchen Tian, Zhengming Zhang, Qinzhen Xu, and Luxi Yang. A privacy-preserved split learning solution for deep learning-based mmwave beam selection. *IEEE Communications Letters*, 2022.

[Wu *et al.*, 2021] Yulei Wu, Hong-Ning Dai, and Haina Tang. Graph neural networks for anomaly detection in industrial internet of things. *IEEE Internet of Things Journal*, 2021.

[Wu *et al.*, 2022] Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. Graph neural networks in recommender systems: a survey. *ACM Computing Surveys*, 55(5):1–37, 2022.

[Xia *et al.*, 2022] Tengxi Xia, Yongheng Deng, Sheng Yue, Junyi He, Ju Ren, and Yaoxue Zhang. Hsfl: An efficient split federated learning framework via hierarchical organization. In *2022 18th International Conference on Network and Service Management (CNSM)*, pages 1–9. IEEE, 2022.

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

[Yang *et al.*, 2022] Ziyuan Yang, Yingyu Chen, Huijie Huangfu, Maosong Ran, Hui Wang, Xiaoxiao Li, and Yi Zhang. Robust split federated learning for u-shaped medical image networks. *arXiv preprint arXiv:2212.06378*, 2022.

[Zhang *et al.*, 2021] Ke Zhang, Carl Yang, Xiaoxiao Li, Lichao Sun, and Siu Ming Yiu. Subgraph federated learning with missing neighbor generation. *Advances in Neural Information Processing Systems*, 34:6671–6682, 2021.

[Zhang *et al.*, 2022a] Miao Zhang, Liangqiong Qu, Praveer Singh, Jayashree Kalpathy-Cramer, and Daniel L Rubin. Splitavg: A heterogeneity-aware federated deep learning method for medical imaging. *IEEE Journal of Biomedical and Health Informatics*, 26(9):4635–4644, 2022.

[Zhang *et al.*, 2022b] Taolin Zhang, Chuan Chen, Yaomin Chang, Lin Shu, and Zibin Zheng. Fedego: Privacy-preserving personalized federated graph learning with ego-graphs. *arXiv preprint arXiv:2208.13685*, 2022.