Unsupervised Federated Graph Learning

Lele Fu[1], **Tianchi Liao**[1], **Sheng Huang**[1], **Bowen Deng**[1], **Chuanfu Zhang**[1],
Shirui Pan[2], **Chuan Chen**[1]*
[1]Sun Yat-sen University, Guangzhou, China
[2]Griffith University, Brisbane, Australia
{fulle,liaotch,huangsh253,dengbw3}@mail2.sysu.edu.cn
s.pan@griffith.edu.au, {zhangchf9,chenchuan}@mail.sysu.edu.cn

Abstract

Federated graph learning (FGL) is a privacy-preserving paradigm for modeling distributed graph data, designed to train a powerful global graph neural network. Existing FGL methods predominantly rely on label information during training, effective FGL in an unsupervised setting remains largely unexplored territory. In this paper, we address two key challenges in unsupervised FGL: 1) Local models tend to converge in divergent directions due to the lack of shared semantic information across clients. Then, how to align representation spaces among multiple clients is the first challenge. 2) Conventional federated weighted aggregation easily results in degrading the performance of the global model, then which raises another challenge, namely how to adaptively learn the global model parameters. In response to the two questions, we propose a tailored framework named FedPAM, which is composed of two modules: Re**p**resentation Sp**a**ce Align**m**ent (RSA) and Ada**p**tive Glob**a**l Para**m**eter Learning (AGPL). RSA leverages a set of learnable anchors to define the global representation space, then local subgraphs are aligned with them through the fused Gromov-Wasserstein optimal transport, achieving the representation space alignment across clients. AGPL stacks local model parameters into third-order tensors, and adaptively integrates the global model parameters in a low-rank tensor space, which facilitates to fuse the high-order knowledge among clients. Extensive experiments on eight graph datasets are conducted, the results demonstrate that the proposed FedPAM is superior over classical and SOTA compared methods.

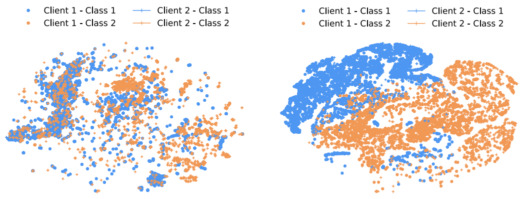
1 Introduction

In response to growing concerns over privacy protection [1, 2, 3], the storage of graph data is becoming increasingly decentralized, where individual clients hold their own private subgraphs. Nevertheless, distributed subgraphs result in the emergence of data silos, triggering data unavailability and impeding the generalization of graph neural networks (GNNs) [4, 5, 6, 7, 8]. In this context, federated graph learning (FGL) [9, 10, 11, 12, 13, 14] has emerged as a paradigm that enables the collaborative training of a powerful GNN across multiple clients, while maintaining the usability of private subgraphs without compromising their confidentiality.

FGL has made substantial progress in recent years. In the context of graph-level tasks, studies [15, 16, 17] primarily concentrate on handling the challenges arising from distributional heterogeneity. For node-level tasks, considerable attentions [18, 9, 19] are devoted to defining the topological heterogeneity in distributed subgraphs and overcoming its adverse effects on the effectiveness of federated training. The existing approaches provide innovative insights and solid solutions for FGL.

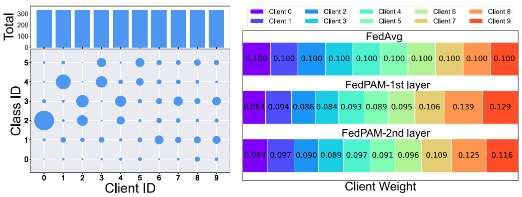
[*] Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).



(a) FedAvg (b) FedPAM

Figure 1: Scatter comparison on Physics dataset for FedAvg and FedPAM. It can be seen that FedPAM achieves a better inter-class separation, aligning the representation spaces of different clients, while FedAvg fails to do so.



(a) Statistics of CiteSeer (b) Comparison of client weights

Figure 2: Comparison of client weights on CiteSeer dataset for FedAvg and FedPAM. FedPAM learns an adaptive weight for each client according to the data heterogeneity while FedAvg assigns a fixed weight for each client based on the number of local samples.

However, they inherently depend on label information during training, indicating that they adhere to supervised or semi-supervised learning paradigms. In practice, data labels are frequently inaccessible, and the lack of consistent semantics across clients further exacerbates the difficulties of federated learning (FL) [20, 21, 22, 23, 24, 25]. As a result, enabling effective FGL in an unsupervised setting poses a significant challenge. For unsupervised FGL, one of the most straightforward solutions is to leverage mature FL frameworks in conjunction with self-supervised learning (SSL) strategies [26, 27, 28, 29]. For example, clients perform local training using the SimCLR [30], while the server aggregates model parameters by FedAvg [31]. However, naively transplanting existing approaches fails to effectively cope with the common issue of heterogeneous data in FL. Moreover, some unsupervised FL algorithms originally designed for non-graph data are also worth mentioning. Studies [32, 33] used the clustering strategies to enhance the consensus of global space. Zhuang et al. [34, 35] proposed to dynamically update the local models based on the parameter differences between the global and local models. Liao et al. [36] addressed the problem of representation collapse in unsupervised FL.

These methods are insightful for unsupervised FL, but they exhibit two primary limitations: ❶ *Inability to align the representation spaces of various clients.* Due to the absence of labels, there are no consistent signals among clients, which can easily lead to shifts in the representation spaces and weaken the generalization for global model. Although studies [32, 36] adopt clustering or optimal transport (OT) to enhance the separability of representations, they merely alleviate representation collapse and do not achieve a unified representation space across clients. Especially, when faced with graph data, their learned representation spaces are prone to fall into suboptimality due to complex topological networks. ❷ *Unable to adaptively learn the optimal parameters of global model.* Most existing federated aggregation methods use weighted averaging based on number of local samples, which might degrade the performance of global model. Studies [34, 35, 36] introduce divergence-aware and multi-objective aggregation strategies, but their matrix-level designs fall short in effectively capturing the complementary information among local models. From Fig. 1(a), we can see that the scatter output by FedAvg is dispersed throughout the space, the nodes of the same categories from various clients cannot be clustered together, indicating that representation spaces of different clients are unaligned. The subgraph statistics for different clients are reported in Fig. 2(a). The total numbers of nodes in the different clients are almost the same, but the numbers of nodes in each category vary considerably, showing different heterogeneous situations. If the traditional FedAvg is used, each client is assigned to almost a same weight as shown in Fig. 2(b), which is clearly unreasonable.

In light of the above concerns, we propose a tailored framework FedPAM for unsupervised FGL, which mainly consists of two key modules: Re**p**resentation Sp**a**ce Align**m**ent (**RSA**) and Ada**p**tive Glob**a**l Para**m**eter Learning (**AGPL**). Specifically, RSA aims to learn a set of cross-client anchors, which are used to define the global representation space. To align the representation spaces of various clients, the fused Gromov-Wasserstein optimal transport (FGW-OT) [37] is employed to establish mapping between local subgraphs and the anchor graph with minimal cost. Additionally, the anchors also serve as global space projector, projecting local embeddings into the global space to facilitate the contrastive learning, thereby mitigating the biased training caused by data heterogeneity. AGPL departs entirely from the conventional weighted aggregation paradigm. It stacks parameters of local models into third-order tensors, and leverages the low-rank tensor decomposition to capture high-order correlations among clients. Further, the optimal parameters of global models are adaptively learned in the low-rank tensor space, enabling effective integration of local knowledge. From Figs. 1(b) and 2(b), it can be observed that the proposed FedPAM can effectively align the representation spaces across clients and adaptively learn the global model parameters. The framework of the proposed FedPAM is presented in Fig. 3. We summarize the contributions of this paper from four-fold:

- We are the first to identify the challenges inherent in unsupervised FGL and design a unified framework to address them. This framework proposes the representation space alignment and the adaptive global parameter learning for unsupervised FGL.
- We learn a set of anchors to span the global representation space, and adopt the FGW-OT to match local subgraphs with the anchor graph, thus achieving the alignment of representation spaces across multiple clients.
- We stack the local model parameters into third-order tensors and extract its low-rank components to capture the high-order correlations among clients. In the high-dimensional space, the optimal global model parameters are obtained by adaptive fusion.
- To verify the effectiveness of the proposed FedPAM, we conduct the comparative experiments on eight graph datasets. Compared to traditional and SOTA methods, FedPAM achieves more superior performance.

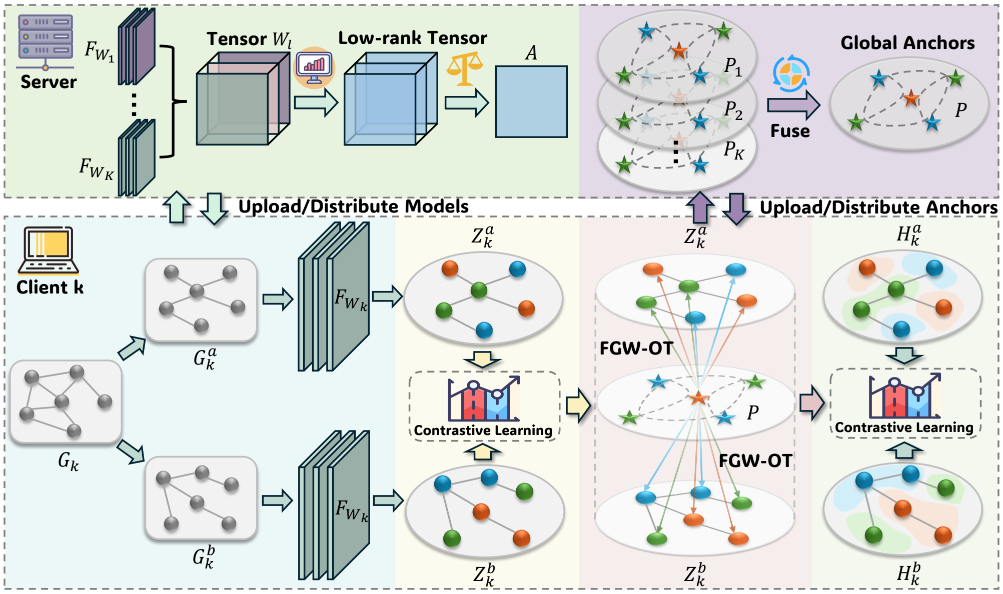


Figure 3: The framework of the proposed FedPAM. The client performs the RSA module and the server performs the AGPL module, models and anchors are transferred between the server and clients.

2 Preliminaries

Federated Graph Learning. Given a FGL system with a central server and K clients, each client possesses a private subgraph $\mathcal{G}_k = \{\mathcal{V}_k, \mathcal{E}_k, \mathbf{X}_k\}$, where $\mathcal{V}_k$, $\mathcal{E}_k$, and $\mathbf{X}_k$ indicate the vector set, the edge set, and the feature matrix, respectively. Specifically, $\mathbf{X}_k \in \mathbb{R}^{N_k \times d}$ has N_k nodes and d dimensions. The adjacency matrix $\mathbf{A}_k \in \mathbb{R}^{N_k \times N_k}$ is induced by the edge set $\mathcal{E}_k$. If there is an edge between the i-th node and the j-th node, then $\mathbf{A}_k(i,j) = 1$. Otherwise, $\mathbf{A}_k(i,j) = 0$. In the unsupervised FGL, the node labels $\mathbf{Y}_k = [y_k^1, ..., y_k^{N_k}]$ are all unavailable during the GNN's training. Notably, we assume that all subgraphs originate from a complete graph, namely $\mathcal{G}_k \in \mathcal{G}$. At the beginning of a communication round in FL, the central server distributes the global GNN $F_{\mathbf{W}}$ parameterized by $\mathbf{W}$ to each client and starts the local training, the computation of a GNN can be summarized as

$$\mathbf{z}_i^{l+1} = \delta\left(F_{\mathbf{W}_l}(\mathbf{z}_i^l, \mathrm{AGG}(\mathbf{z}_j^l, \mathbf{A}_{ij}))\right) |\forall v_j \in \mathcal{N}(v_i)), \tag{1}$$

where $\delta(\cdot)$ denotes the activation function, $F_{\mathbf{W}_l}$ denotes the l-th encoder layer, $\mathbf{z}_i^{l+1}$ is the i-th node's representation at the $(l+1)$-th layer, $\mathcal{N}(v_i)$ is the neighborhood node set for node v_i, $\mathrm{AGG}(\cdot)$ denotes the embedding fusion manner. When the local training finishes, local GNNs are uploaded to the server for aggregating the global GNN, which is formulated as

$$\mathbf{W} = \sum_{k=1}^{K} \frac{N_k}{N} \mathbf{W}_k, \tag{2}$$

where $N = \sum_{k=1}^{K} N_k$ denotes the sum of local nodes.

Tensor Nuclear Norm. For a third-order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, its the j-th frontal slice is denoted as $\boldsymbol{\mathcal{X}}^{(j)} \in \mathbb{R}^{n_1 \times n_2}$. The fast Fourier transformation (FFT) is used to transform a tensor into the frequency domain, which is formulated as $\hat{\boldsymbol{\mathcal{X}}} = \mathrm{fft}(\boldsymbol{\mathcal{X}}, [], 3)$, its inverse operation is written as $\boldsymbol{\mathcal{X}} = \mathrm{ifft}(\hat{\boldsymbol{\mathcal{X}}}, [], 3)$. The product $\boldsymbol{\mathcal{Z}} \in \mathbb{R}^{n_1 \times n_4 \times n_3}$ of $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{n_2 \times n_4 \times n_3}$ is defined as

$$\boldsymbol{\mathcal{Z}} = \boldsymbol{\mathcal{X}} * \boldsymbol{\mathcal{Y}} = \mathrm{fold}(\mathrm{bcirc}(\boldsymbol{\mathcal{X}})\, \mathrm{bvec}(\boldsymbol{\mathcal{Y}})), \tag{3}$$

where $\mathrm{bvec}(\boldsymbol{\mathcal{Y}}) = [\boldsymbol{\mathcal{Y}}^{(1)}; ...; \boldsymbol{\mathcal{Y}}^{(n_3)}]$ is the block vectorizing operation, and $\mathrm{fold}(\mathrm{bvec}(\boldsymbol{\mathcal{Y}}))$ is the inverse operation. $\mathrm{bcirc}(\boldsymbol{\mathcal{X}})$ is defined as

$$bcirc(\boldsymbol{\mathcal{X}}) = \begin{bmatrix} \boldsymbol{\mathcal{X}}^{(1)} & \boldsymbol{\mathcal{X}}^{(n_3)} & \cdots & \boldsymbol{\mathcal{X}}^{(2)} \\ \boldsymbol{\mathcal{X}}^{(2)} & \boldsymbol{\mathcal{X}}^{(1)} & \cdots & \boldsymbol{\mathcal{X}}^{(3)} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\mathcal{X}}^{(n_3)} & \boldsymbol{\mathcal{X}}^{(n_3-1)} & \cdots & \boldsymbol{\mathcal{X}}^{(1)} \end{bmatrix}. \tag{4}$$

Further, if $\boldsymbol{\mathcal{X}}^T * \boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{X}} * \boldsymbol{\mathcal{X}}^T = \boldsymbol{\mathcal{I}}$, $\boldsymbol{\mathcal{X}}$ is called orthogonal tensor. $\boldsymbol{\mathcal{I}} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$ is an identity tensor, where the first frontal slice is an identity matrix and the remaining slices are all zeros. The tensor singular value decomposition (T-SVD) is a key operation for calculating the tensor nuclear norm, their definitions are introduced as follows.

Definition 1. *For a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the T-SVD is formulated as*

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{U}} * \boldsymbol{\mathcal{D}} * \boldsymbol{\mathcal{V}}^T \tag{5}$$

where $\boldsymbol{\mathcal{U}} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$ and $\boldsymbol{\mathcal{V}} \in \mathbb{R}^{n_2 \times n_2 \times n_3}$ are orthogonal, $\boldsymbol{\mathcal{D}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is f-diagonal.

Definition 2. *For a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, its T-SVD based tensor nuclear norm [38] is defined as*

$$||\boldsymbol{\mathcal{X}}||_{\circledast} = \sum_{j=1}^{n_3} ||\hat{\boldsymbol{\mathcal{X}}}^{(j)}||_* = \sum_{i}^{\min(n_1,n_2)} \sum_{j=1}^{n_3} |\hat{\boldsymbol{\mathcal{D}}}^{(j)}(i,i)|, \tag{6}$$

where $\hat{\boldsymbol{\mathcal{D}}}^{(j)}(i,i)$ is obtained by $\hat{\boldsymbol{\mathcal{X}}}^{(j)} = \hat{\boldsymbol{\mathcal{U}}}^{(j)} * \hat{\boldsymbol{\mathcal{D}}}^{(j)} * \hat{\boldsymbol{\mathcal{V}}}^{(j)^T}$.

3 Methodology

In this section, we elaborate on the proposed FedPAM. Overall, FedPAM is comprised of two modules: **RSA** and **AGPL**. Specifically, **RSA** is used to align the representation spaces across clients through the FGW-OT between local subgraphs and anchor graph. **AGPL** is employed to adaptively learn the global model parameters in the low-rank tensor space.

3.1 Fused Gromov-Wasserstein Optimal Transport for Aligning Representation Spaces

Local Self-Supervised Learning. Self-supervised learning (SSL) is an effective technique for preventing representation collapse and enhancing the discriminative quality of learned representations. In the absence of node labels, local clients adopt a self-supervised strategy for training. For the k-th local subgraph $\mathcal{G}_k$, it is augmented to $\mathcal{G}_k^a$ and $\mathcal{G}_k^b$ via two different kinds of augment methods. Then, the local GNN $F_{\mathbf{W}_k}^k$ encodes them into the embedding space:

$$\mathbf{Z}_k^a = F_{\mathbf{W}_k}^k(\mathcal{G}_k^a), \mathbf{Z}_k^b = F_{\mathbf{W}_k}^k(\mathcal{G}_k^b). \tag{7}$$

A SSL loss between $\mathbf{Z}_k^a$ and $\mathbf{Z}_k^b$ is used for back propagation and generally written as

$$\mathcal{L}_{SSL} = \mathbb{E}_{\forall i,j \in [N_k]} \ell_{pos}(\mathbf{z}_i^a, \mathbf{z}_i^b) - \lambda \ell_{neg}(\mathbf{z}_i^a, \{\mathbf{z}_j^b\}_{i \neq j}), \tag{8}$$

where ℓ_{pos} and ℓ_{neg} denote the losses for positive and negative pairs, respectively. Overall, the SSL loss $\mathcal{L}_{SSL}$ can be categorized into two classes: contrastive and non-contrastive. Contrastive methods require the involvement of negative samples during the computation of SSL, while non-contrastive methods do not rely on negative samples. We investigate the impacts of various SSL on the proposed framework in the experimental section.

FGW-OT between Local Graphs and Anchor Graph. Under our assumption, local subgraphs originate from a underlying global graph, implying that node representations across clients following a unified distribution. However, due to the heterogeneous nature of local data, the embeddings learned by local clients tend to exhibit distributional discrepancies. Moreover, since data from other clients is unaccessible, local models lack awareness of the global distribution. To address this issue, we introduce N_P learnable anchors $\mathbf{P} = \{\mathbf{p}_1, ..., \mathbf{p}_{N_P}\}$ in the server to span the global space, the relationships between anchors are recorded by the anchor graph $\mathbf{A}_P$. These anchors also serve as the global projector, enabling the projection of local embedding into the global space. At the beginning of each communication round, the global model and the anchors are distributed to each client for local training. To align local subgraphs with anchor graph, the FGW-OT is introduced, whose definition is written as

Definition 3. *Given two metric measure space $\mathcal{G}_1 = (\mathcal{V}_1, \mathbf{A}_1, \mu_1)$ and $\mathcal{G}_1 = (\mathcal{V}_2, \mathbf{A}_2, \mu_2)$, $\mathcal{V}_1$ and $\mathcal{V}_2$ denote the node sets, where the data sizes are respectively N_1 and N_2, $\mathbf{A}_1$ and $\mathbf{A}_2$ denote the relationship matrices, μ_1 and μ_2 denote the data marginal distributions. Then, the FGW-OT distance between $\mathcal{G}_1$ and $\mathcal{G}_2$ is defined as*

$$\begin{aligned}&\inf_{\pi\in\Pi}\sum_{i,j}^{N_1}\sum_{m,n}^{N_2}\left((1-\alpha)D(\mathbf{x}_{1,i},\mathbf{x}_{2,j})^p + \alpha|[\mathbf{A}_1]_{i,j} - [\mathbf{A}_2]_{m,n}|^p\right)\pi_{i,m}\pi_{j,n}\\ &\textit{s.t. } \Pi = \{\pi \in \mathbb{R}_+^{N_1\times N_2} | \pi\mathbf{1}_{N_2} = \mu_1, \pi^T\mathbf{1}_{N_1} = \mu_2\}\end{aligned} \tag{9}$$

where $\mathbf{x}_{1,i} \in \mathcal{V}_1$, $\mathbf{x}_{2,j} \in \mathcal{V}_2$, $D(\cdot)$ measures the distance between two nodes, α is a trade-off parameter between the feature-level distance and the relationship-level distance, p is a constant used to adjust the strength of the distance, $\mathbf{1}_{N_1}$ and $\mathbf{1}_{N_2}$ denote the two vectors of all-one elements with N_1 and N_2 dimensions. $\pi \in \mathbb{R}^{N_1\times N_2}$ is the expected optimal transport plan.

In context of graph learning, FGW-OT can measure the sum of feature and topology distances between two graphs. Since anchors are derived from the global representation space, thus the anchor graph can be optimally transported to each local subgraph $\mathcal{G}_k$ with minimal cost. The optimization objective is formulated as

$$\begin{aligned}&\min_{\pi_k\in\Pi}\sum_{i,j}^{N_k}\sum_{m,n}^{N_P}\left((1-\alpha)(-[\mathbf{P}_k]_m[\mathbf{Z}_k^T]_i) + \alpha\,|[\mathbf{A}_k]_{i,j} - [\mathbf{A}_P]_{m,n}|\right)[\pi_k]_{i,m}[\pi_k]_{j,n} - \epsilon\,\mathrm{H}(\pi_k),\\ &\text{s.t. } \Pi = \{\pi_k \in \mathbb{R}_+^{N_k\times N_P} | \pi_k\mathbf{1}_{N_P} = \mu_1, \pi_k^T\mathbf{1}_{N_k} = \mu_2\},\end{aligned} \tag{10}$$

where $\mathrm{H}(\cdot)$ denotes the entropy of certain variable, ϵ denotes a trade-off constant. The entropy regularization is introduced to ensure that the optimization problem has a unique solution. The product $\mathbf{P}_k\mathbf{Z}_k^T$ measures the similarity between the anchors and nodes, then $-\mathbf{P}_k\mathbf{Z}_k^T$ is the distance between the two terms. Problem (10) can be optimized by the proposed method in [37], then an optimal transport matrix $\pi_k \in \mathbb{R}^{N_k\times N_P}$ is obtained. π_k reflects the unnormalized transfer probability from nodes to anchors, and its normalized form is formulated as

$$[\mathbf{T}_k]_{i,j} = \frac{[\pi_k]_{i,j}}{\sum_{j'}[\pi_k]_{i,j'}}. \tag{11}$$

For the anchor-node similarity matrix $\mathbf{P}_k\mathbf{Z}_k^T$, its has two-fold meanings. **First**, it depicts the relationships between anchors and nodes, and its normalized form can also be viewed as their transfer probability. Thus, we have

$$[\mathbf{C}_k]_{i,j} = \frac{\exp([\mathbf{Z}_k]_i[\mathbf{P}_k^T]_j/\tau)}{\sum_{j'}\exp([\mathbf{Z}_k]_i[\mathbf{P}_k^T]_{j'}/\tau)}. \tag{12}$$

$\mathbf{T}_k$ is closed-form solution obtained through algebraic computation and does not have gradients, whereas $\mathbf{C}_k$ is an inexact solution with gradients. In addition, $\mathbf{C}_k$ is also used to conduct the anchor graph $\mathbf{A}_P = \mathbf{C}_k^T \mathbf{C}_k$. $\mathbf{C}_k$ is expected to drive close to $\mathbf{T}_k$, then the loss is written as:

$$\mathcal{L}_{CE} = -\sum_{i=1}^{N_k} \sum_{j=1}^{N_P} \left([\mathbf{T}_k]_{i,j} \log [\mathbf{C}_k]_{i,j}\right). \tag{13}$$

Notably, as mentioned above, there are two augmented latent embeddings: $\mathbf{Z}_k^a$ and $\mathbf{Z}_k^b$. We perform the OT mapping on the two views, then $\mathcal{L}_{CE}$ is reformulated as

$$\mathcal{L}_{CE} = \mathcal{L}_{CE}^a + \mathcal{L}_{CE}^b. \tag{14}$$

Second, the anchor matrix $\mathbf{P}$ represents the entire representation space and is regarded as a global space projector. From another perspective, $\mathbf{P}\mathbf{Z}^T$ is understood as a global-aware representation. In Eq. (8), it performs the contrastive learning with local representations and is denoted as $\mathcal{L}_{SSL}^l$. In addition, we expect to the global-aware representations to be more discriminative as well. By projecting $\mathbf{Z}_k^a$ and $\mathbf{Z}_k^b$ into global space via anchors $\mathbf{P}_k$, the global-aware representations $\mathbf{H}_k^a = \mathbf{Z}_k^a \mathbf{P}_k^T$ and $\mathbf{H}_k^b = \mathbf{Z}_k^b \mathbf{P}_k^T$ are obtained, then the contrastive learning between them is executed through Eq. (8), the loss is denoted as $\mathcal{L}_{SSL}^g$. Thus, the local SSL loss is reformulated as

$$\mathcal{L}_{SSL} = \mathcal{L}_{SSL}^l + \mathcal{L}_{SSL}^g. \tag{15}$$

In summary, the training loss in local clients is concluded as

$$\mathcal{L} = \mathcal{L}_{SSL} + \lambda \mathcal{L}_{CE}, \tag{16}$$

where λ denotes the trade-off parameter. Through the back propagation from loss $\mathcal{L}$, the local GNNs and anchors are optimized. When the local training completes, the local GNNs and anchors are uploaded to sever for aggregation.

3.2 Low-Rank Tensor Optimization for Adaptively Learning Global GNN Parameters

Traditional federated aggregation performs weighted parameter fusion based on number of local samples. However, this approach has two drawbacks. First, the matrix-level fusion cannot capture the high-order correlations between clients. Second, the fixed weights cannot adaptively determine the importance of each client. In light of the shortcomings, we propose to find the optimal global GNN's parameters in a low-rank tensor space. Concretely, when the server receives local GNNs' parameters, the l-th layer's parameters $\{\mathbf{W}_{k,l}\}_{k=1}^K$ are stacked as a third-order tensor $\boldsymbol{\mathcal{W}}_l$. We argue that the low-rank component $\boldsymbol{\mathcal{L}}_l$ in $\boldsymbol{\mathcal{W}}_l$ contains the globally shared information, and the noise component $\boldsymbol{\mathcal{E}}_l$ contains the client-specific information. Then, the former needs to be retained, while the latter should be discarded. Further, we aim to adaptively fuse local GNNs' parameters in the low-rank tensor space for obtaining the optimal global GNN parameters $\mathbf{A}_l$. Considering the above concerns, the following optimization objective is proposed (the subscript l is omitted for a concise expression):

$$\begin{aligned} &\min_{\boldsymbol{\mathcal{L}}, \boldsymbol{\mathcal{E}}, \mathbf{A}, \alpha_k} \|\boldsymbol{\mathcal{L}}\|_{\circledast} + \beta \|\boldsymbol{\mathcal{E}}\|_1 + \sum_{k=1}^K (\alpha_k)^r \left(\|\mathbf{L}_k - \mathbf{A}\|_F^2\right) \\ &\text{s.t. } \boldsymbol{\mathcal{W}} = \boldsymbol{\mathcal{L}} + \boldsymbol{\mathcal{E}}, \sum_{k=1}^K \alpha_k = 1, \alpha_k \geq 0, \forall k \in [K], \end{aligned} \tag{17}$$

where β is a trade-off parameter, α_k is the weight for the k-th client, r is a constant to smooth the distribution of weights. The optimization problem (17) can be solved by the alternating direction method of multipliers (ADMM). The augmented Lagrangian function of (17) is conducted as follows:

$$\begin{aligned} &\mathcal{F}\left(\boldsymbol{\mathcal{L}}; \boldsymbol{\mathcal{E}}; \boldsymbol{\mathcal{G}}; \mathbf{A}; \{\alpha_k\}_{k=1}^K\right) \\ &= \|\boldsymbol{\mathcal{G}}\|_{\circledast} + \beta \|\boldsymbol{\mathcal{E}}\|_1 + \sum_{k=1}^K (\alpha_k)^r \|\mathbf{L}_k - \mathbf{A}\|_F^2 + \langle \boldsymbol{\mathcal{Y}}, \boldsymbol{\mathcal{W}} - \boldsymbol{\mathcal{L}} - \boldsymbol{\mathcal{E}} \rangle + \frac{\mu}{2} \|\boldsymbol{\mathcal{W}} - \boldsymbol{\mathcal{L}} - \boldsymbol{\mathcal{E}}\|_F^2 \\ &+ \langle \boldsymbol{\mathcal{H}}, \boldsymbol{\mathcal{G}} - \boldsymbol{\mathcal{L}} \rangle + \frac{\mu}{2} \|\boldsymbol{\mathcal{G}} - \boldsymbol{\mathcal{L}}\|_F^2, \end{aligned} \tag{18}$$

where $\boldsymbol{\mathcal{G}}$ is an auxiliary variable, $\boldsymbol{\mathcal{Y}}$ and $\boldsymbol{\mathcal{H}}$ denote the two Lagrange multipliers, μ is a trade-off parameter. The update rule for each variable is presented as follows.

Update $\boldsymbol{\mathcal{G}}$: Fixing $\boldsymbol{\mathcal{L}}$, $\boldsymbol{\mathcal{E}}$, $\mathbf{A}$, $\{\alpha_k\}_{k=1}^K$, the subproblem with respect to $\boldsymbol{\mathcal{G}}$ is written as

$$\begin{aligned}&\min_{\boldsymbol{\mathcal{G}}} \|\boldsymbol{\mathcal{G}}\|_{\circledast} + \langle \boldsymbol{\mathcal{H}}, \boldsymbol{\mathcal{G}} - \boldsymbol{\mathcal{L}} \rangle + \frac{\mu}{2}\|\boldsymbol{\mathcal{G}} - \boldsymbol{\mathcal{L}}\|_F^2 \\ &= \min_{\boldsymbol{\mathcal{G}}} \|\boldsymbol{\mathcal{G}}\|_{\circledast} + \frac{\mu}{2}\left\|\boldsymbol{\mathcal{G}} - \left(\boldsymbol{\mathcal{L}} - \frac{1}{\mu}\boldsymbol{\mathcal{H}}\right)\right\|_F^2 .\end{aligned} \tag{19}$$

For the solution of low-rank tensor $\boldsymbol{\mathcal{G}}$, study [39] provided a tensor tubal-shrinkage method $\mathcal{R}$, the updated $\boldsymbol{\mathcal{G}}^*$ is obtained by

$$\boldsymbol{\mathcal{G}}^* = \mathcal{R}_{1/\mu}(\boldsymbol{\mathcal{L}} - \frac{1}{\mu}\boldsymbol{\mathcal{H}}). \tag{20}$$

Update $\boldsymbol{\mathcal{L}}$: When optimizing $\boldsymbol{\mathcal{L}}$, the update rule for its each slice $\mathbf{L}_k$ is same. Fixing $\boldsymbol{\mathcal{G}}$, $\boldsymbol{\mathcal{E}}$, $\mathbf{A}$, $\{\alpha_k\}_{k=1}^K$, the subproblem for $\mathbf{L}_k$ is formulated as

$$\begin{aligned}\min_{\mathbf{L}_k} (\alpha_k)^r \|\mathbf{L}_k - \mathbf{A}\|_F^2 + \langle \mathbf{Y}_k, \mathbf{W}_k - \mathbf{L}_k - \mathbf{E}_k \rangle + \frac{\mu}{2}\|\mathbf{W}_k - \mathbf{L}_k - \mathbf{E}_k\|_F^2 \\ + \langle \mathbf{H}_k, \mathbf{G}_k - \mathbf{L}_k \rangle + \frac{\mu}{2}\|\mathbf{G}_k - \mathbf{L}_k\|_F^2 .\end{aligned} \tag{21}$$

Taking the partial derivative with respect to $\mathbf{L}_k$ and setting it to zero, the update rule for $\mathbf{L}_k^*$ is derived:

$$\mathbf{L}_k^* = \frac{2(\alpha_k)^r\mathbf{A} + \mu\mathbf{Z}_k - \mu\mathbf{E}_k + \mathbf{Y}_k + \mu\mathbf{G}_k + \mathbf{H}_k}{2(\alpha_k)^r + 2\mu} \tag{22}$$

Update $\mathbf{A}$: Masking variables unrelated to $\mathbf{A}$, we obtain the following subproblem:

$$\min_{\mathbf{A}} \sum_{k=1}^K (\alpha_k)^r(||\mathbf{L}_k - \mathbf{A}||_F^2). \tag{23}$$

The update rule for $\mathbf{A}^*$ can be obtained by computing the partial derivative with respect to $\mathbf{A}^*$ and equating it to zero, then we have

$$\mathbf{A}^* = \frac{\sum_{k=1}^K (\alpha_k)^r\mathbf{L}_k}{\sum_{k=1}^K (\alpha_k)^r}. \tag{24}$$

Update $\boldsymbol{\mathcal{E}}$: When $\boldsymbol{\mathcal{E}}$ is updated, the other variables are viewed as constants, the subproblem is formulated as

$$\begin{aligned}&\min_{\boldsymbol{\mathcal{E}}} \beta||\boldsymbol{\mathcal{E}}||_1 + \langle \boldsymbol{\mathcal{Y}}, \boldsymbol{\mathcal{W}} - \boldsymbol{\mathcal{L}} - \boldsymbol{\mathcal{E}} \rangle + \frac{\mu}{2}||\boldsymbol{\mathcal{W}} - \boldsymbol{\mathcal{L}} - \boldsymbol{\mathcal{E}}||_F^2 \\ &= \min_{\boldsymbol{\mathcal{E}}} \beta||\boldsymbol{\mathcal{E}}||_1 + \frac{\mu}{2}||\boldsymbol{\mathcal{E}} - (\boldsymbol{\mathcal{W}} - \boldsymbol{\mathcal{L}} + \boldsymbol{\mathcal{Y}}/\mu)||_F^2.\end{aligned} \tag{25}$$

Let $\boldsymbol{\mathcal{B}} = \boldsymbol{\mathcal{W}} - \boldsymbol{\mathcal{L}} + \boldsymbol{\mathcal{Y}}/\mu$, the update method with respect to $\boldsymbol{\mathcal{E}}$ is

$$\boldsymbol{\mathcal{E}}^* = \max(\boldsymbol{\mathcal{B}} - \lambda/\mu, 0) + \min(\boldsymbol{\mathcal{B}} + \lambda/\mu, 0). \tag{26}$$

Update α_k: Neglecting variables unrelated to α_k, the subproblem about α_k is written as

$$\min_{\alpha_k} \sum_{k=1}^K (\alpha_k)^r(||\mathbf{L}_k - \mathbf{A}||_F^2), \text{s.t.} \sum_{k=1}^K \alpha_k = 1, \alpha_k \geq 0, \forall k \in [K]. \tag{27}$$

Taking the partial derivative with respect to α_k and setting it to zero, we have

$$\alpha_k^* = \frac{(||\mathbf{L}_k - \mathbf{A}||_F^2)^{1/(1-r)}}{\sum_{k=1}^K (||\mathbf{L}_k - \mathbf{A}||_F^2)^{1/(1-r)}}. \tag{28}$$

Update $\boldsymbol{\mathcal{Y}}$, $\boldsymbol{\mathcal{H}}$, μ:

$$\boldsymbol{\mathcal{H}}^* = \boldsymbol{\mathcal{H}} + \mu(\boldsymbol{\mathcal{G}} - \boldsymbol{\mathcal{L}}); \boldsymbol{\mathcal{Y}}^* = \boldsymbol{\mathcal{Y}} + \mu(\boldsymbol{\mathcal{W}} - \boldsymbol{\mathcal{L}} - \boldsymbol{\mathcal{E}}); \mu^* = \min(\omega * \mu, \mu_{max}), \tag{29}$$

where ω and μ_{max} are two predefined parameters. The optimal global GNN parameters are obtained through the iterative optimization based on ADMM described above. Thus, the global GNN parameters and anchors are distributed to clients in the next communication.

Table 1: Performance comparison on Cora, CiteSeer, PubMed, and Ogbn-Arxiv datasets with three different SSL losses, where the optimal results are **bolded** and the suboptimal results are underlined.

SSL	Method	Cora		CiteSeer		PubMed		Ogbn-Arxiv	
		ACC	Fscore	ACC	Fscore	ACC	Fscore	ACC	Fscore
Simsiam	FedAvg	54.38	39.27	36.30	20.34	63.81	50.88	35.26	20.49
	FedProx	55.35	40.83	36.80	21.34	63.81	50.88	34.61	20.03
	MOON	54.03	39.81	35.94	20.41	63.80	50.87	35.46	21.20
	FedU^2	56.04	42.65	39.00	25.77	61.27	48.09	35.39	22.00
	FedPAM	**61.40**	**52.85**	**49.53**	**38.73**	**64.23**	**51.92**	**35.97**	**23.07**
SimCLR	FedAvg	54.65	39.95	35.48	19.62	64.08	51.46	45.58	33.08
	FedProx	55.26	40.68	35.56	19.78	63.95	51.17	44.40	31.86
	MOON	54.56	39.81	36.30	20.34	64.42	52.17	46.23	33.96
	FedX	56.73	43.11	37.10	22.37	61.77	49.09	46.69	34.77
	FedU^2	57.09	43.45	40.89	27.54	66.79	56.25	**48.41**	36.46
	FedPAM	**58.62**	**45.21**	**43.90**	**31.61**	**69.22**	**59.42**	48.19	**37.25**
BYOL	FedAvg	63.01	50.51	42.65	29.72	61.85	49.62	35.29	20.73
	FedU	64.06	51.92	46.81	34.98	62.08	49.82	38.07	24.15
	FedEMA	63.53	51.23	43.59	30.97	64.47	53.39	38.12	24.07
	Orchestra	54.47	39.63	37.54	24.39	61.27	48.09	35.26	20.49
	FedU^2	65.11	54.16	47.32	35.79	66.51	55.86	39.95	26.03
	FedPAM	**66.01**	**55.23**	**48.81**	**38.69**	**66.67**	**56.25**	**40.55**	**26.79**

4 Experiments

4.1 Experimental Setups

Graph Datasets. Eight graph datasets are selected as benchmark datasets, including **Cora**, **CiteSeer**, **PubMed**, **Ogbn-Arxiv**, **Computers**, **Photo**, **Physics**, **Amazon-ratings**. The eight datasets vary in terms of scales, and cover different types, such as citation network, co-purchase network, co-author network, rating network. Each graph is divided into ten subgraphs via the Louvain method [40].

Compared Methods. We compare the proposed FedPAM with typical and SOTA FL algorithms, including **FedAvg** [31], **FedProx** [41], **MOON** [42], **FedX** [43], **FedU** [34], **FedEMA** [35], **Orchestra** [32], **FedU^2** [36]. In local training, three SSL strategies are adopted, including **Simsiam** [44], **SimCLR** [30], **BYOL** [45]. Specifically, **Simsiam** and **BYOL** are non-contrastive methods while **SimCLR** is contrastive method. We adopt the standard linear probing to evaluate the performance of algorithms, ACC and Fscore are used as the evaluation metrics.

4.2 Performance Comparison

We evaluate the performance of FL algorithms using three SSL strategies on eight graph datasets, the results are reported in Tables 1 and 2. Some interesting phenomenon can be observed. First, FL algorithms exhibit varying performance when equipped with different SSL losses. For instance, on Cora dataset, employing BYOL leads to improvements by 8.63% and 8.36% in ACC compared to the other two SSL losses for FedAvg, respectively. Encouragingly, the proposed FedPAM consistently achieves superior performance with different SSL methods, validating the contributions of RSA and AGPL. Second, compared to baseline methods, the proposed FedPAM demonstrates more significant improvements on homogeneous graphs than on heterogeneous graphs. Node attributes and edge types in homogeneous graphs are less diverse, then the learnable anchors are more likely to span the global representation space. Furthermore, homogeneous graphs induce local models to be more similar, facilitating low-rank tensor optimization to find better global model parameters.

4.3 Ablation Study

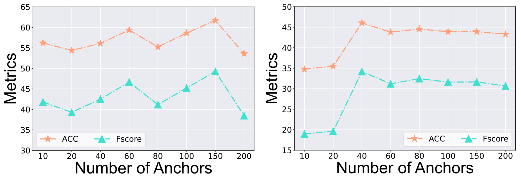
RSA and AGPL paly critical roles for the proposed FedPAM. The results of their ablation studies are reported in Table 3. When both modules are removed, the model degenerates into FedAvg and

Table 2: Performance comparison on Computers, Photo, Physics, and Amazon-ratings datasets with three different SSL losses, where the optimal results are **bolded** and the suboptimal results are <u>underlined</u>.

SSL	Method	Computers		Photo		Physics		Amazon-ratings	
		ACC	Fscore	ACC	Fscore	ACC	Fscore	ACC	Fscore
Simsiam	FedAvg	63.09	50.92	72.01	63.56	75.55	67.73	<u>36.72</u>	<u>19.79</u>
	FedProx	<u>64.05</u>	<u>51.70</u>	<u>73.26</u>	<u>65.08</u>	75.33	66.22	36.64	19.83
	MOON	63.92	52.16	72.46	64.20	77.82	70.79	36.69	19.86
	FedU^2	63.20	51.03	72.65	64.57	<u>86.26</u>	<u>81.02</u>	<u>36.72</u>	<u>19.79</u>
	FedPAM	**64.19**	**53.93**	**74.06**	**66.96**	**86.51**	**81.86**	**37.45**	**24.97**
SimCLR	FedAvg	78.21	72.27	84.59	79.82	79.07	71.30	36.56	20.06
	FedProx	78.41	72.56	84.66	79.82	77.05	67.74	36.46	19.92
	MOON	78.90	72.53	84.72	79.96	80.23	73.10	36.72	20.05
	FedX	<u>80.60</u>	<u>75.09</u>	<u>85.49</u>	<u>80.72</u>	80.31	73.60	36.70	19.79
	FedU^2	79.24	73.34	84.94	80.15	<u>81.03</u>	<u>74.27</u>	<u>39.58</u>	<u>27.92</u>
	FedPAM	**81.63**	**77.42**	**85.78**	**81.25**	**83.76**	**78.16**	**39.60**	**29.69**
BYOL	FedAvg	63.38	52.20	71.12	61.93	84.99	79.13	36.77	21.05
	FedU	65.00	54.92	74.19	67.42	85.36	79.54	37.11	24.07
	FedEMA	65.44	<u>56.38</u>	<u>76.81</u>	<u>70.31</u>	83.39	77.53	38.32	**27.22**
	Orchestra	62.84	52.67	73.74	67.22	72.16	62.10	36.81	20.12
	FedU^2	<u>65.55</u>	54.73	74.86	68.11	<u>85.41</u>	<u>79.74</u>	<u>36.82</u>	20.50
	FedPAM	**66.66**	**57.57**	**80.50**	**75.34**	**86.43**	**80.98**	**38.43**	<u>26.52</u>

Table 3: Ablation study with respect to two key modules **RSA** and **AGPL** on Cora and CiteSeer datasets, where SimCLR loss is used in the local training.

RSA	AGPL	Cora		CiteSeer	
		ACC	Fscore	ACC	Fscore
✗	✗	54.65	39.95	35.48	19.62
✗	✓	56.65	44.18	<u>41.35</u>	<u>27.80</u>
✓	✗	<u>58.03</u>	**45.27**	37.89	22.65
✓	✓	**58.62**	<u>45.21</u>	**43.90**	**31.61**

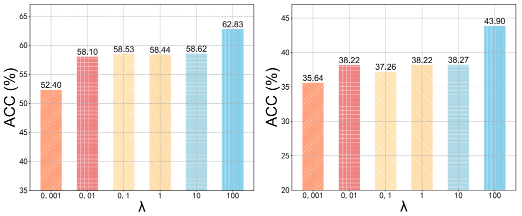


(a) Cora

(b) CiteSeer

Figure 4: Performance comparison with different numbers of anchors on Cora and CiteSeer datasets, where SimCLR loss is used in the local training.

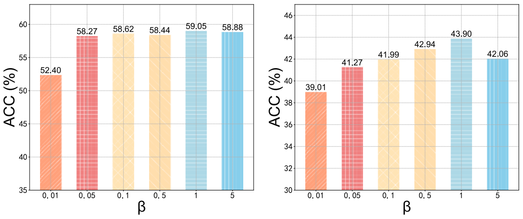
naturally achieves the worst performance. When either RSA and AGPL module is present, the performance of the model is improved, indicating that both representation space alignment and adaptive global parameter learning are effective. Certainly, the performance is optimal with both modules available.



(a) Cora

(b) CiteSeer

Figure 5: Sensitivity study with respect to λ on Cora and CiteSeer datasets, where SimCLR loss is used in the local training.



(a) Cora

(b) CiteSeer

Figure 6: Sensitivity study with respect to β on Cora and CiteSeer datasets, where SimCLR loss is used in the local training.

4.4 Parameter Sensitivity Investigation

We investigate the impacts of three key hyperparameters in the proposed FedPAM, including the number of anchors M, λ in the local training, and β in the low-rank tensor optimization. From Fig. 4, it can be seen that either too few or too many anchors is detrimental to achieving optimal performance. Excessive anchors tend to reduce discriminability, while too few anchors may lead to representation collapse. λ controls the strength of the optimal transport, a large value of λ facilitates better alignment of representation spaces as shown in Fig. 5. However, setting β too high is not advisable as shown in Fig. 6, because it may hinder the capture of client-specific information.

5 Conclusion

In this paper, we address two key challenges in unsupervised FGL: the misalignment of representation spaces and non-adaptive federated aggregation schemas. For the first challenge, we propose the RSA module, which aims to learn a set of anchors across clients and use them to align representation spaces of different clients. For the second challenge, the AGPL module is introduced, employing the low-rank tensor optimization to adaptively learn a global model. Finally, the experimental results on eight graph datasets verify the effectiveness of the proposed FedPAM. More details about related work, algorithm, dadasets, and experimental results can be referred in the Appendix.

Acknowledgments

The research is supported by the National Key R&D Program of China (2023YFB2703700), the National Natural Science Foundation of China (62176269).

References

[1] Qinbin Li, Zeyi Wen, Zhaomin Wu, Sixu Hu, Naibo Wang, Yuan Li, Xu Liu, and Bingsheng He. A survey on federated learning systems: Vision, hype and reality for data privacy and protection. *IEEE Transactions on Knowledge and Data Engineering*, 35(4):3347–3366, 2023.

[2] Wenke Huang, Mang Ye, Zekun Shi, Guancheng Wan, He Li, Bo Du, and Qiang Yang. Federated learning for generalization, robustness, fairness: A survey and benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):9387–9406, 2024.

[3] Yichen Li, Haozhao Wang, Wenchao Xu, Tianzhe Xiao, Hong Liu, Minzhu Tu, Yuying Wang, Xin Yang, Rui Zhang, Shui Yu, Song Guo, and Ruixuan Li. Unleashing the power of continual learning on non-centralized devices: A survey. *IEEE Communications Surveys & Tutorials*, 2025.

[4] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24, 2020.

[5] Bowen Deng, Tong Wang, Lele Fu, Sheng Huang, Chuan Chen, and Tao Zhang. Thesaurus: Contrastive graph clustering by swapping fused gromov-wasserstein couplings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 16199–16207, 2025.

[6] Bowen Deng, Lele Fu, Jialong Chen, Sheng Huang, Tianchi Liao, Zhang Tao, and Chuan Chen. Towards understanding parametric generalized category discovery on graphs. In *Proceedings of International Conference on Machine Learning*, 2025.

[7] Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*, 36(7):3580–3599, 2024.

[8] Gabriele Corso, Hannes Stark, Stefanie Jegelka, Tommi Jaakkola, and Regina Barzilay. Graph neural networks. *Nature Reviews Methods Primers*, 4(1):17, 2024.

[9] Wenke Huang, Guancheng Wan, Mang Ye, and Bo Du. Federated graph semantic and structural learning. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 3830–3838, 2023.

[10] Lele Fu, Bowen Deng, Sheng Huang, Tianchi Liao, Shirui Pan, and Chuan Chen. Less is more: Federated graph learning with alleviating topology heterogeneity from a causal perspective. In *Proceedings of International Conference on Machine Learning*, 2025.

[11] Rui Liu, Pengwei Xing, Zichao Deng, Anran Li, Cuntai Guan, and Han Yu. Federated graph neural networks: Overview, techniques, and challenges. *IEEE Transactions on Neural Networks and Learning Systems*, 36(3):4279–4295, 2025.

[12] Guancheng Wan, Zitong Shi, Wenke Huang, Guibin Zhang, Dacheng Tao, and Mang Ye. Energy-based backdoor defense against federated graph learning. In *Proceedings of International Conference on Learning Representations*, 2025.

[13] Jingxin Liu, Jieren Cheng, Renda Han, Wenxuan Tu, Jiaxin Wang, and Xin Peng. Federated graph-level clustering network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 18870–18878, 2025.

[14] Sheng Huang, Lele Fu, Tianchi Liao, Bowen Deng, Chuanfu Zhang, and Chuan Chen. Fedbg: Proactively mitigating bias in cross-domain graph federated learning using background data. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 5408–5416, 2025.

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

[16] Yue Tan, Yixin Liu, Guodong Long, Jing Jiang, Qinghua Lu, and Chengqi Zhang. Federated learning on non-iid graphs via structural knowledge sharing. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 9953–9961, 2023.

[17] Zihan Tan, Guancheng Wan, Wenke Huang, and Mang Ye. Fedssp: Federated graph learning with spectral knowledge and personalized preference. In *Advances in Neural Information Processing Systems*, 2024.

[18] Ke Zhang, Carl Yang, Xiaoxiao Li, Lichao Sun, and Siu Ming Yiu. Subgraph federated learning with missing neighbor generation. *Proceedings of the Advances in Neural Information Processing Systems*, pages 6671–6682, 2021.

[19] Guancheng Wan, Wenke Huang, and Mang Ye. Federated graph learning under domain shift with generalizable prototypes. In *Proceedings of the AAAI conference on artificial intelligence*, pages 15429–15437, 2024.

[20] Jinyu Cai, Yunhe Zhang, Jicong Fan, and See-Kiong Ng. Lg-fgad: An effective federated graph anomaly detection framework. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 3760–3769, 2024.

[21] Zhuang Qi, Lei Meng, Zhaochuan Li, Han Hu, and Xiangxu Meng. Cross-silo feature space alignment for federated learning on clients with imbalanced data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 19986–19994, 2025.

[22] Lei Meng, Zhuang Qi, Lei Wu, Xiaoyu Du, Zhaochuan Li, Lizhen Cui, and Xiangxu Meng. Improving global generalization and local personalization for federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, 36(1):76–87, 2025.

[23] Ming Hu, Peiheng Zhou, Zhihao Yue, Zhiwei Ling, Yihao Huang, Anran Li, Yang Liu, Xiang Lian, and Mingsong Chen. Fedcross: Towards accurate federated learning via multi-model cross-aggregation. In *IEEE International Conference on Data Engineering*, pages 2137–2150, 2024.

[24] Ming Hu, Zhihao Yue, Xiaofei Xie, Cheng Chen, Yihao Huang, Xian Wei, Xiang Lian, Yang Liu, and Mingsong Chen. Is aggregation the only choice? federated learning via layer-wise model recombination. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1096–1107, 2024.

[25] Yichen Li, Haozhao Wang, Yining Qi, Wei Liu, and Ruixuan Li. Re-fed+: A better replay strategy for federated incremental learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(7):5489–5500, 2025.

[26] Yixin Liu, Ming Jin, Shirui Pan, Chuan Zhou, Yu Zheng, Feng Xia, and S Yu Philip. Graph self-supervised learning: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 35(6):5879–5900, 2022.

[27] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):857–876, 2023.

[28] Danielle L Ferreira, Connor Lau, Zaynaf Salaymang, and Rima Arnaout. Self-supervised learning for label-free segmentation in cardiac ultrasound. *Nature Communications*, 16(1):4070, 2025.

[29] Guancheng Wan, Yijun Tian, Wenke Huang, Nitesh V Chawla, and Mang Ye. S3GCL: Spectral, swift, spatial graph contrastive learning. In *Proceedings of International Conference on Machine Learning*, 2024.

[30] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PmLR, 2020.

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

[32] Ekdeep Lubana, Chi Ian Tang, Fahim Kawsar, Robert Dick, and Akhil Mathur. Orchestra: Unsupervised federated learning via globally consistent clustering. In *Proceedings of International Conference on Machine Learning*, pages 14461–14484, 2022.

[33] Yawen Wu, Zhepeng Wang, Dewen Zeng, Meng Li, Yiyu Shi, and Jingtong Hu. Decentralized unsupervised learning of visual representations. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, pages 2326–2333, 2022.

[34] Weiming Zhuang, Xin Gan, Yonggang Wen, Shuai Zhang, and Shuai Yi. Collaborative unsupervised visual representation learning from decentralized data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4912–4921, 2021.

[35] Weiming Zhuang, Yonggang Wen, and Shuai Zhang. Divergence-aware federated self-supervised learning. In *Proceedings of International Conference on Learning Representations*, 2022.

[36] Xinting Liao, Weiming Liu, Chaochao Chen, Pengyang Zhou, Fengyuan Yu, Huabin Zhu, Binhui Yao, Tao Wang, Xiaolin Zheng, and Yanchao Tan. Rethinking the representation in federated unsupervised learning with non-iid data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22841–22850, 2024.

[37] Vayer Titouan, Nicolas Courty, Romain Tavenard, and Rémi Flamary. Optimal transport for structured data with application on graphs. In *Proceedings of International Conference on Machine Learning*, pages 6275–6284, 2019.

[38] Misha E Kilmer, Karen Braman, Ning Hao, and Randy C Hoover. Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging. *SIAM Journal on Matrix Analysis and Applications*, 34(1):148–172, 2013.

[39] Wenrui Hu, Dacheng Tao, Wensheng Zhang, Yuan Xie, and Yehui Yang. The twist tensor nuclear norm for video completion. *IEEE transactions on neural networks and learning systems*, 28(12):2961–2973, 2016.

[40] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

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

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

[43] Sungwon Han, Sungwon Park, Fangzhao Wu, Sundong Kim, Chuhan Wu, Xing Xie, and Meeyoung Cha. Fedx: Unsupervised federated learning with cross knowledge distillation. In *Proceedings of European Conference on Computer Vision*, pages 691–707, 2022.

[44] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021.

[45] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020.

[46] Changxin Tian, Yuexiang Xie, Xu Chen, Yaliang Li, and Xin Zhao. Privacy-preserving cross-domain recommendation with federated graph learning. *ACM Transactions on Information Systems*, 42(5):1–29, 2024.

[47] Xunkai Li, Zhengyu Wu, Wentao Zhang, Yinlin Zhu, Rong-Hua Li, and Guoren Wang. Fedgta: Topology-aware averaging for federated graph learning. *Proceedings of the VLDB Endowment*, pages 41–50, 2023.

[48] Yinlin Zhu, Xunkai Li, Zhengyu Wu, Di Wu, Miao Hu, and Rong-Hua Li. Fedtad: Topology-aware data-free knowledge distillation for subgraph federated learning. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 2024.

[49] Eduardo Fernandes Montesuma, Fred Maurice Ngolè Mboula, and Antoine Souloumiac. Recent advances in optimal transport for machine learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(2):1161–1180, 2025.

[50] Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(9):1853–1865, 2016.

[51] Igor Melnyk, Youssef Mroueh, Brian Belgodere, Mattia Rigotti, Apoorva Nitsure, Mikhail Yurochkin, Kristjan Greenewald, Jiri Navratil, and Jarret Ross. Distributional preference alignment of llms via optimal transport. *Advances in Neural Information Processing Systems*, pages 104412–104442, 2024.

[52] Liqun Chen, Zhe Gan, Yu Cheng, Linjie Li, Lawrence Carin, and Jingjing Liu. Graph optimal transport for cross-domain alignment. In *Proceedings of International Conference on Machine Learning*, pages 1542–1553, 2020.

[53] Yuguang Yan, Canlin Yang, Yuanlin Chen, Ruichu Cai, and Michael Ng. Hypergraph learning for unsupervised graph alignment via optimal transport. *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 21913–21921, 2025.

[54] Pan Zhou, Canyi Lu, Zhouchen Lin, and Chao Zhang. Tensor factorization for low-rank tensor completion. *IEEE Transactions on Image Processing*, 27(3):1152–1163, 2017.

[55] Lele Fu, Zhaoliang Chen, Yongyong Chen, and Shiping Wang. Unified low-rank tensor learning and spectral embedding for multi-view subspace clustering. *IEEE Transactions on Multimedia*, 25:4972–4985, 2022.

[56] Zixiao Yu, Lele Fu, Yongyong Chen, Zhiling Cai, and Guoqing Chao. Hyper-laplacian regularized concept factorization in low-rank tensor space for multi-view clustering. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 9(2):1728–1742, 2025.

[57] Vishwanath Saragadam, Randall Balestriero, Ashok Veeraraghavan, and Richard G. Baraniuk. Deeptensor: Low-rank tensor decomposition with deep network priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):10337–10348, 2024.

[58] Renwei Dian, Yuanye Liu, and Shutao Li. Spectral super-resolution via deep low-rank tensor representation. *IEEE Transactions on Neural Networks and Learning Systems*, 36(3):5140–5150, 2025.

[59] Yejin Kim, Jimeng Sun, Hwanjo Yu, and Xiaoqian Jiang. Federated tensor factorization for computational phenotyping. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 887–895, 2017.

[60] Tianchi Liao, Lele Fu, Lei Zhang, Lei Yang, Chuan Chen, Michael K. Ng, Huawei Huang, and Zibin Zheng. Privacy-preserving vertical federated learning with tensor decomposition for data missing features. *IEEE Transactions on Information Forensics and Security*, 20:3445–3460, 2025.

[61] Lele Fu, Sheng Huang, Yuecheng Li, Chuan Chen, Chuanfu Zhang, and Zibin Zheng. Learn the global prompt in the low-rank tensor space for heterogeneous federated learning. *Neural Networks*, 187:107319, 2025.

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

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: In this paper, we address the two key challenges of unsupervised federated graph learning, and propose Representation Space Alignment module and Adaptive Global Parameter Learning module. To our knowledge, we are the first to recognize and address the challenges of unsupervised federated graph learning.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: In the appendix, we discuss the computational complexity of the proposed method and point out the limitations of excessive computational complexity, which is the direction of our future improvement.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: This paper does not contain theoretical innovations. However, we are the first to recognize the challenges of unsupervised federated graphs and propose a novel framework for solving them.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We introduce the experimental environment and parameter settings in the implementation details.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The algorithm flow, used datasets, and the parameter settings are introduced in details.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We present methods for subgraph partitioning, hyperparameter settings, and used optimizer in the paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We report the ACC and Fscore of the different algorithms on the used datasets, which are sufficient indicators of the performance gap between the different algorithms.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We report the configuration of the server used in the implementation details.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research conforms to the NIPS Code of Ethics in all respects.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The proposed method aims to improve the effectiveness of unsupervised federated graph learning, which has a positive effect on protecting users' privacy and improving the training efficiency.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have cited sources for the compared algorithms and datasets used in the paper.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: We guarantee that the proposed FedPAM and the paper are completely original and have not been generated with the help of LLM.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Summary of Appendix

We present the following sections in the appendix as supplement to this manuscript:

- Related work about FGL, OT, and low-rank tensor optimization.
- The algorithm flow of the proposed FedPAM.
- Implementation details for the proposed FedPAM.
- The analysis for the computational complexity.
- The details of the eight graph datasets used.
- The performance comparison using KNN as the evaluation metric.
- The performance comparison with large-scale clients.
- The performance comparison of plugging the AGPL module into existing works.

B Related Work

B.1 Federated Graph Learning

The objective of federated graph learning is to enable the collaborative training of a GNN on multiple decentralized graphs, thereby improving its generalization. In terms of training data, FGL is generally divided into two categories: graph-level and node-level. Graph-level FGL focuses on addressing challenges related to label or feature heterogeneity. For example, [15, 16] proposed to cluster clients or share topology patterns for Non-IID graphs. To mitigate the domain gaps between private graphs, [46, 17] captured the cross-domain knowledge to enhance the local personalized inference. In contrast, node-level FGL primarily targets challenges arising from topological heterogeneity. [18, 9, 19] leveraged the global information to assist the local training. [47, 48] emphasized the impact of topological structures on federated learning and proposed the topology-aware aggregation method. While the aforementioned approaches provide valuable insights, they overlook the problem of FGL in unsupervised scenarios. We recognize the challenges of unsupervised FGL and propose a novel approach to address them.

B.2 Optimal Transport

Optimal transport [49] is used to find the shortest distance for transforming one distribution into another, and has been widely applied in the fields of domain adaption and domain generalization. [50, 51] aligned the distributions between the source domain and the target domain via OT. As an extension of classical optimal transport, GW-OT is designed to align structural relationships between data distributions, making it particularly suitable for applications in graph match tasks. For instance, [52, 53] aimed to match the topological structures between various graphs via GW-OT. Furthermore, to simultaneously align the distributions in terms of features and structures, the FGW-OT is proposed. [37, 5] adopted the FGW-OT to achieve optimal mapping from source graphs to target graphs through features and topologies. For unsupervised FGL, how to explore the global representation space is critical challenge, we leverage the FGW-OT to learn a group of anchors across multiple private subgraphs for positioning the global coordinate.

B.3 Low-rank Tensor Decomposition

Low-rank tensor decomposition seeks to recover the underlying low-rank components from a given tensor, and it has found extensive applications in image restoration, data mining, and data compression tasks. [54, 55, 56] employed the tensor singular value decomposition to explore the principal information of multi-source data. Deep learning provides a new solution for exploring the low-rankness, [57, 58] used neural networks to nonlinearly capture the low-rank structures. Although there have been some existing studies working on federated tensor computation [59, 60, 61], they mainly focused on distributed tensor decomposition with respect to data features. In contrast, we leverage low-rank tensor decomposition to exploit the complementarity of local models and solve for the global model parameters in the low-rank tensor space.

C The Algorithm Flow of the Proposed FedPAM

Algorithm 1 The main steps of FedPAM

Input: Number of clients K, communication rounds T, local training epochs E, learning rate η, trade-off parameters λ and β, local subgraph $\mathcal{G}_k = (\mathcal{V}_k, \mathcal{E}_k, \mathbf{X}_k)$, local GNN $F_{\mathbf{W}_k}$.

Output: Global model $F_{\mathbf{W}}$ and Global anchors $\mathbf{P}$.

```
Client Side:
for $k = 1 : K$ in parallel do
    for epoch $e = 1 : E$ do
        Augment the original local subgraph $\mathcal{G}_k$ into $\mathcal{G}_k^a$ and $\mathcal{G}_k^b$ via two different methods;
        Encode the two views of subgraphs: $\mathbf{Z}_k^a \leftarrow F_{\mathbf{W}_k}^k(\mathcal{G}_k^a)$, $\mathbf{Z}_k^b \leftarrow F_{\mathbf{W}_k}^k(\mathcal{G}_k^b)$;
        Calculate the SSL loss for local representations: $\mathcal{L}_{SSL}^l \leftarrow (\mathbf{Z}_k^a, \mathbf{Z}_k^b)$ by Eq. (8);
        for view $i$ in $(a, b)$ do
            Calculate the optimal transport matrix $\mathbf{T}_k^i$ between $\mathbf{Z}_k^i$ and $\mathbf{P}_k$ by Algorithm 2;
            Calculate the normalized transport matrix: $\mathbf{T}_k^i \leftarrow (\pi_k^i)$ by Eq. (11);
            Calculate the transfer probability matrix: $\mathbf{C}_k^i \leftarrow (\mathbf{Z}_k^i, \mathbf{P}_k)$ by Eq. (12);
            Calculate the CE loss $\mathcal{L}_{CE}^i \leftarrow (\mathbf{T}_k^i, \mathbf{C}_k^i)$ by (13);
        end for
        Calculate the CE loss for two views: $\mathcal{L}_{CE} \leftarrow (\mathcal{L}_{CE}^a, \mathcal{L}_{CE}^b)$;
        Obtain the global-aware representations: $\mathbf{H}_k^a \leftarrow \mathbf{Z}_k^a \mathbf{P}_k^T$, $\mathbf{H}_k^b \leftarrow \mathbf{Z}_k^b \mathbf{P}_k^T$;
        Calculate the SSL loss for global-aware representations: $\mathcal{L}_{SSL}^g \leftarrow (\mathbf{H}_k^a, \mathbf{H}_k^b)$ by Eq. (8);
        Calculate the total SSL loss $\mathcal{L}_{SSL} \leftarrow (\mathcal{L}_{SSL}^l, \mathcal{L}_{SSL}^g)$ by Eq. (15);
        Update local GNN $\mathbf{W}_k^e \leftarrow \mathbf{W}_k^{e-1} - \eta \nabla (\mathcal{L}_{SSL} + \mathcal{L}_{CE})$;
        Update local anchors $\mathbf{P}_k^e \leftarrow \mathbf{P}_k^{e-1} - \eta \nabla (\mathcal{L}_{SSL} + \mathcal{L}_{CE})$;
    end for
    Upload the local GNN $F_{\mathbf{W}_k}^k$ and local anchors $\mathbf{P}_k$ to the server;
end for
Server Side:
for $t = 1 : T$ do
    for $i = 1 : L$ do
        Stack the $l$-th layer's parameters $\{\mathbf{W}_{k,l}\}$ of local model into a third-order tensor $\boldsymbol{\mathcal{W}}_l$;
        Obtain the optimal parameters $\mathbf{A}_l$ from $\boldsymbol{\mathcal{W}}_l$ by Algorithm 4;
    end for
    Obtain the global anchors $\mathbf{P} = \sum_{k=1}^{K} \frac{N_k}{N} \mathbf{P}_k$;
    Distribute the global GNN $F_{\mathbf{W}}$ and the global anchors $\mathbf{P}$ to clients;
end for
```

Algorithm 2 The main steps of fused GW-OT

Input: Node marginal distribution $\nu \in \mathbb{R}^N$, anchor marginal distribution $\mu \in \mathbb{R}^M$, adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$, anchor adjacency matrix $\mathbf{S} \in \mathbb{R}^{M \times M}$, the transport matrix $\varphi \in \mathbb{R}^{N \times M}$ in terms of feature level, trade-off parameters α and ϵ, maximum iteration round T.

Output: The transport matrix $\pi \in \mathbb{R}^{N \times M}$.

```
$\pi \leftarrow \nu \mu^T$;
$G_1 \leftarrow \mathbf{S} \odot \mathbf{S} \cdot \nu \mathbb{1}^T$;
$G_2 \leftarrow \mathbb{1}(\mathbf{A} \odot \mathbf{A} \cdot \mu)^T$;
for $t = 1 : T$ do
    $G \leftarrow 2\alpha \cdot (-2\mathbf{S}\pi\mathbf{A} + G_1 + G_2) + (1 - \alpha)\varphi$;
    $\pi \leftarrow \mathrm{OT}(\nu, \mu, G, \pi, \epsilon)$ via Algorithm 3;
end for
```

D Implement Details

A 2-layer graph convolutional network [62] is used as the backbone, and the latent embedding dimension is set as 128. Following [18, 48], the Louvain method [40] is used to divide the initial

Algorithm 3 The main steps of OT via Sinkhorn algorithm

```
Input: Node marginal distribution $\nu \in \mathbb{R}^N$, anchor marginal distribution $\mu \in \mathbb{R}^M$, cost matrix $\mathbf{C}^{N \times M}$, initialized transport matrix $\pi$, trade-off parameter $\epsilon$, maximum iteration round $T^{'}$.
Output: The transport matrix $\pi \in \mathbb{R}^{N \times M}$.
$K \leftarrow e^{-\mathbf{C}/\epsilon}$;
for $t = 1 : T^{'}$ do
    $\nu \leftarrow \nu / K\mu$;
    $\mu \leftarrow \mu / K\nu$;
end for
$\pi \leftarrow \mathrm{diag}(\nu) K \, \mathrm{diag}(\mu)$;
```

Algorithm 4 The main steps for low-rank tensor optimization

```
Input: Observed tensor $\boldsymbol{\mathcal{W}}$, trade-off parameters $\beta$ and $\mu$, step $\omega$, maximum $\mu_{max} = 10^{10}$, threshold $\epsilon$. Initial the auxiliary variable $\boldsymbol{\mathcal{G}} = \mathbf{0}$, the Lagrange multipliers $\boldsymbol{\mathcal{H}} = \mathbf{0}$, $\boldsymbol{\mathcal{Y}} = \mathbf{0}$.
Output: Consistent matrix $\mathbf{A}$.
while not convergent do
    Update $\boldsymbol{\mathcal{G}}^{t+1} \leftarrow (\boldsymbol{\mathcal{L}}^t, \boldsymbol{\mathcal{H}}^t)$ by Eq. (20);
    for $i = 1 : K$ do
        Update $\mathbf{L}_k^{t+1} \leftarrow (\mathbf{A}^t, \mathbf{Z}_k^t, \mathbf{E}_k^t, \mathbf{Y}_k^t, \mathbf{G}_k^t, \mathbf{H}_k^t, \alpha_k^t, \mu^t)$ by Eq. (22);
    end for
    Update $\mathbf{A}^{t+1} \leftarrow (\mathbf{L}_k^t, \alpha_k^t)$ by Eq. (24);
    Update $\boldsymbol{\mathcal{E}}^{t+1} \leftarrow (\boldsymbol{\mathcal{B}}^t, \beta/\mu^t)$ by Eq. (26);
    Update $\alpha_k^{t+1} \leftarrow (\mathbf{L}_k^t, \mathbf{A}^t)$ by Eq. (28);
    Check the convergence conditions:
    $\max(||\boldsymbol{\mathcal{W}}^{t+1} - \boldsymbol{\mathcal{L}}^{t+1} - \boldsymbol{\mathcal{E}}^{t+1}||_F^2, ||\boldsymbol{\mathcal{L}}^{t+1} - \boldsymbol{\mathcal{L}}^t||_F^2, ||\boldsymbol{\mathcal{E}}^{t+1} - \boldsymbol{\mathcal{E}}^t||_F^2) \leq \epsilon$
    ▷ When the conditions are satisfied, the iteration ends, otherwise it continues;
    $t = t + 1$;
end while
```

graph to multiple local subgraphs, the number of clients is set as 10. We use Adam as the optimizer with the learning rate set to 0.001. The numbers of communication round and local training rounds are set to 100 and 5, respectively. When the federated training completes, we use the standard linear probing to evaluate the performance of algorithms. The values of ACC and Fscore on test sets are reported. Some trade-off parameters are required to be predefined, α and ϵ in FGW-OT is fixed as 0.5 and 1. λ is tuned in $\{0.1, 5, 10, 50, 70, 100\}$, β is varied in $\{0.1, 0.5, 1\}$, the number of anchors is tuned in $\{30, 60, 100, 500, 1000\}$.

E The Analysis for the Computational Complexity

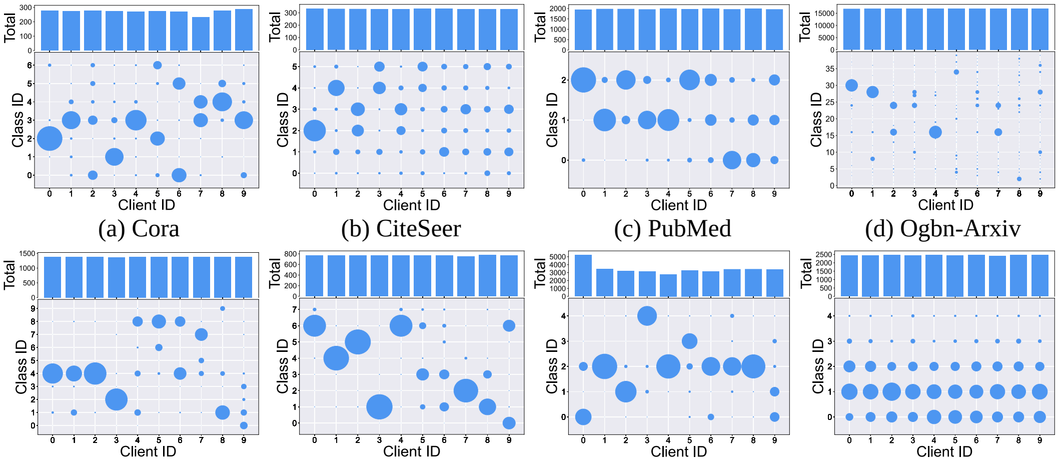
The computation complexity mainly originates from two aspects: the FGW-OT for aligning representation spaces and the low-rank tensor optimization for adaptively learning global model parameters. For the FGW-OT, the computation of G in Algorithm 2 takes $\mathcal{O}(N_E M + M^2 N)$, where N_E denotes the number of edges, M denotes the number of anchors, and N denotes the number of nodes. Notably, the adjacency matrices are often sparse, so the computational complexity can be significantly decreased. Likewise, the OT based on Sinkhorn algorithm in Algorithm 3 takes $\mathcal{O}(MN)$. Then, the FGW-OT takes $\mathcal{O}(N_E M + M^2 N + MN)$. For the low-rank tensor optimization, the update of $\boldsymbol{\mathcal{G}}$ with $D_{l,1} \times D_{l,2} \times K$ in Algorithm 4 occupies the dominant computational complexity. First, the FFT and the inverse FFT require $\mathcal{O}(D_{l,1}^2 D_{l,2} \log(K))$, where $D_{l,1}$ and $D_{l,2}$ are the dimensions for the l-th layer's parameters, K is the number of clients, and $D_{l,1}$ is the higher dimension. Second, the T-SVD costs $\mathcal{O}(D_{l,1} D_{l,2}^2 K)$. Then, the low-rank tensor optimization takes $\mathcal{O}(D_{l,1}^2 D_{l,2} \log(K) + D_{l,1} D_{l,2}^2 K)$. Overall, the computational complexity for the proposed FedPAM is $\mathcal{O}(N_E M + M^2 N + MN + D_{l,1}^2 D_{l,2} \log(K) + D_{l,1} D_{l,2}^2 K)$.

F The Details of Eight Graph Datasets

We conduct the experiments on eight graph datasets, including Cora, CiteSeer, PubMed, Ogbn-Arxiv, Computers, Photo, Physics, Amazon-ratings. Their detailed information about the numbers of nodes, features, edges, classes, the proportions of dataset divisions, and the categories of datasets are described in Table 4. The statistical information of local subgraphs for eight graph datasets is presented in Fig. 7.

Table 4: Descriptions of eight graph datasets.

Dataset	Nodes	Features	Edges	Classes	Train / Val / Test	Category
Cora	2,708	1,433	5,429	7	20% / 40% / 40%	Citation Network
CiteSeer	3,327	3,703	4,732	6	20% / 40% / 40%	Citation Network
PubMed	19,717	500	44,338	3	20% / 40% / 40%	Citation Network
Ogbn-Arxiv	169,343	128	231,559	40	60% / 20% / 20%	Citation Network
Computers	13,381	767	245,778	10	20% / 40% / 40%	Co-purchase Network
Photo	7,487	745	119,043	8	20% / 40% / 40%	Co-purchase Network
Physics	34,493	8,415	247,962	5	20% / 40% / 40%	Co-author Network
Amazon-ratings	24,492	300	93,050	5	50% / 25% / 25%	Rating Network



(a) Cora (b) CiteSeer (c) PubMed (d) Ogbn-Arxiv

(e) Computers (f) Photo (g) Physics (h) Amazon-ratings

Figure 7: Statistical information of local subgraphs for eight graph datasets when the number of clients is fixed as 10.

G The Performance Comparison Using KNN as the Evaluation Metric

Standard linear probing is sensitive to linearly differentiable features and may not accurately evaluate the model if the data distribution is complex. In addition to using the standard linear probing, we use KNN as the evaluation metric, the results are reported in Table 5. When using Simsiam and BOYL as the SSL losses on CiteSeer dataset, the values of evaluation metrics are much better than that when using standard linear probing. Hence, employing different evaluation methods provide a more comprehensive understanding of models. Fortunately, it can be seen that the proposed FedPAM achieves optimal performance in most cases, proving that embedding output by FedPAM has better discriminative property.

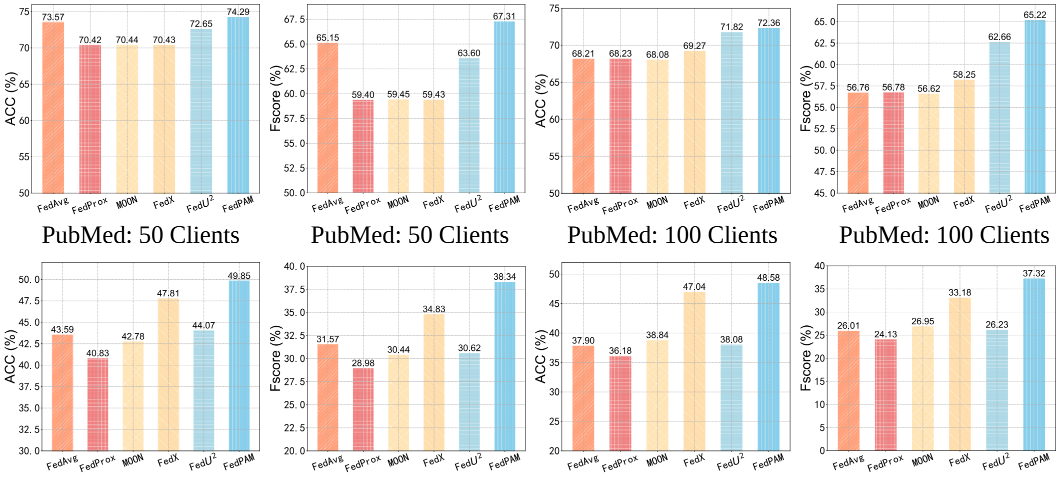
H The Performance Comparison with Large-Scale Clients

Performance validation on large-scale clients is an important measure for the scalability of FL methods, we test the performance of various FL methods with 50 and 100 clients on PubMed and Ogbn-Arxiv (OA) datasets, the results are presented in Fig. 8, where SimCLR is used as the SSL loss.

Table 5: Performance comparison on Cora and CiteSeer datasets with three different SSL losses. The KNN is used as evaluation method, where the optimal results are **bolded** and the suboptimal results are underlined.

SSL	Method	Cora		CiteSeer	
		ACC	Fscore	ACC	Fscore
Simsiam	FedAvg	54.55	42.90	36.66	25.71
	FedProx	55.17	41.80	39.59	28.01
	MOON	56.28	44.83	36.75	26.75
	FedU^2	54.90	41.92	40.99	30.44
	FedPAM	**58.93**	**47.27**	**50.40**	**38.01**
SimCLR	FedAvg	52.49	38.05	37.77	24.60
	FedProx	53.28	39.10	36.37	22.73
	MOON	53.01	38.74	37.68	24.80
	FedX	54.31	40.62	**42.74**	**31.16**
	FedU^2	54.30	40.68	38.78	25.78
	FedPAM	**57.71**	**44.53**	42.07	30.43
BYOL	FedAvg	64.74	**53.63**	44.88	34.17
	FedU	63.35	52.59	46.25	35.33
	FedEMA	64.20	52.53	43.30	30.62
	Orchestra	54.12	41.25	36.89	24.65
	FedU^2	59.87	48.18	44.58	33.33
	FedPAM	**64.99**	53.08	**49.25**	**39.68**

It can be seen that the proposed FedPAM stays ahead of the curve even with a large number of clients, showing that representation space alignment and adaptive global parameter learning are effective. Notably, the superiority of the proposed FedPAM is more obvious in terms of Fscore, indicating that the proposed FedPAM not only has a high overall prediction accuracy, but also shows a better performance on the minority classes.



PubMed: 50 Clients PubMed: 50 Clients PubMed: 100 Clients PubMed: 100 Clients

OA: 50 Clients OA: 50 Clients OA: 100 Clients OA: 100 Clients

Figure 8: Performance comparison with more clients on PubMed and Ogbn-Arxiv (OA) datasets, where the number of clients is set to 50 and 100, respectively.

I The Performance Comparison of Plugging the AGPL Module into Existing Works

Notably, the proposed AGPL module is a plug-and-play module, which can be incorporated into existing works and promote their performance. Specifically, we plug AGPL into FedAvg, FedProx,

Table 6: Performance comparison on Cora, CiteSeer, and Ogbn-Arxiv datasets when the AGPL module is plugged into existing works, where the optimal results are **bolded**.

SSL	Method	Cora		CiteSeer		Ogbn-Arxiv	
		ACC	Fscore	ACC	Fscore	ACC	Fscore
Simsiam	FedAvg	54.38	39.27	36.30	20.34	35.26	20.49
	FedAvg+AGPL	**58.56**	**46.67**	**43.29**	**30.60**	**35.33**	**21.21**
	FedProx	55.35	40.83	36.80	21.34	34.61	20.03
	FedProx+AGPL	**60.15**	**47.07**	**43.75**	**31.58**	**35.26**	**20.49**
	MOON	54.03	39.81	35.94	20.41	35.46	21.20
	MOON+AGPL	**58.92**	**46.22**	**41.34**	**30.42**	**35.44**	**21.59**
SimCLR	FedAvg	54.65	39.95	35.48	19.62	45.58	33.08
	FedAvg+AGPL	**56.65**	**44.18**	**41.35**	**27.80**	**47.10**	**37.13**
	FedProx	55.26	40.68	35.56	19.78	44.40	31.86
	FedProx+AGPL	**56.79**	**46.29**	**41.54**	**28.61**	**47.36**	**35.01**
	MOON	54.56	39.81	36.30	20.34	46.23	33.96
	MOON+AGPL	**56.49**	**43.74**	**41.82**	**29.26**	**47.91**	**39.04**
	FedX	56.73	43.11	37.10	22.37	46.69	34.77
	FedX+AGPL	**57.97**	**44.56**	**42.92**	**30.75**	**48.76**	**39.56**

MOON, and FedX with Simsiam and SimCLR losses, respectively. Table 6 reports the experimental results. It can be observed that the performance of existing works after equipped with AGPL is improved, demonstrating its effectiveness.