

# Federated Graph-Level Clustering Network

Jingxin Liu<sup>1</sup>, Jieren Cheng<sup>2,3</sup>, Renda Han<sup>2</sup>, Wenxuan Tu<sup>2,3</sup>, Jiaxin Wang<sup>2</sup>, Xin Peng<sup>4</sup>

<sup>1</sup>School of Cyberspace Security, Hainan University, Haikou, China

<sup>2</sup>School of Computer Science and Technology, Hainan University, Haikou, China

<sup>3</sup>Hainan Blockchain Technology Engineering Research Center, Haikou, China

<sup>4</sup>School of Computer, National University of Defense Technology, Changsha, China  
{ljx\_0417, cjr22, wenxuantu}@163.com

## Abstract

Federated graph learning (FGL), which excels in analyzing non-IID graphs as well as protecting data privacy, has recently emerged as a hot topic. Existing FGL methods usually train the client model using labeled data and then collaboratively learn a global model without sharing their local graph data. However, in real-world scenarios, the lack of data annotations impedes the negotiation of multi-source information at the server, leading to sub-optimal feedback to the clients. To address this issue, we propose a novel unsupervised learning framework called **F**ederated **G**raph-level **C**lustering **N**etwork (FedGCN), which collects the topology-oriented features of non-IID graphs from clients to generate global consensus representations through multi-source clustering structure sharing. Specifically, in the client, we first preserve the prototype features of each cluster from the structure-oriented embedding through clustering and then upload the learned multiple prototypes that are hard to be reconstructed into the raw graph data. In the server, we generate consensus prototypes from multiple condensed structure-oriented signals through Gaussian estimation, which are subsequently transferred to each client to promote the great encoding capacity of the local model for better clustering. Extensive experiments across multiple non-IID graph datasets have demonstrated the effectiveness and superiority of FedGCN against its competitors.

## Introduction

Federated graph learning (FGL) (Huang et al. 2024; Wan, Huang, and Ye 2024; Huang et al. 2024) has emerged as a prominent research direction in graph machine learning in recent years. Its key advantage is that it enables distributed learning on non-IID graph data through collaborative training without sharing the raw data. FGL methods have been widely applied to various graph analysis tasks, including graph-level classification (Tan et al. 2023), link prediction (Hu et al. 2022), and node classification (Cheung, Dai, and Li 2021).

The main idea of most current FGL methods is to utilize annotated graph data to learn high-quality representations within each client for effective collaborative learning at the server. Despite their great successes, the assumption of available labeled graphs may not always hold in practical scenarios since it is time-consuming and cost-expensive

to acquire enough annotations for large amounts of graph data. To alleviate this issue, recent federated multi-view graph learning methods first allocate a single view with a fixed topology to each client and subsequently integrate uploaded multi-source sample representations or cluster assignments into a unified consensus for improved clustering performance (Yan, Wang, and Jin 2024; Ren et al. 2024). However, the above studies have one of the least following non-negligible limitations: they 1) are not tailored for learning cross-domain graphs, which are commonly encountered in practical applications (Tan et al. 2023); 2) suffer from privacy leakage for graph data, as the attribute information uploaded to the server can be easily reconstructed into its original form (Fredrikson, Jha, and Ristenpart 2015). This factor may potentially increase the risk of violating the federated learning principle. Consequently, it is crucial to develop a novel unsupervised FGL paradigm that can not only flexibly handle the cross-domain non-IID graphs without annotations but also has a more effective and safer collaborative learning capacity to protect graph data privacy.

An intuitive solution is to allocate different graph datasets to each client and upload multi-source information that is difficult to mutually reconstruct. Following this principle, we make one step toward unsupervised federated graph learning with cross-domain datasets, specifically focusing on federated graph-level clustering. To fulfill this, there are two key challenges to be addressed, i.e., 1) the information that is well-protected for privacy within graphs needs to be collected and preserved in the unsupervised local model; 2) it is difficult to capture consensus characteristics from cross-domain graphs to boost the quality of collaborative learning. For the first challenge, inspired by the previous work (Tan et al. 2023), we propose that extracting and uploading the structure-oriented information not only effectively protects privacy but also safely reflects the proximity of sample correlations. For the second challenge, we assume that the features learned by deep neural networks can be approximated with a mixture of Gaussian distribution (Luo et al. 2021; Chen et al. 2020). With this assumption, it has the potential to generate consensus graph embeddings by exploiting the clustering hints from multiple structure-oriented information.

Based on the above observations, we propose a novel unsupervised FGL framework for both cross-dataset and cross-

domain graph data, termed Federated Graph-level Clustering Network (FedGCN). The key idea of our approach is to collect topology-oriented features of client-side non-IID graphs and generate global consensus graph embeddings by sharing the multi-source clustering structure. Specifically, on the client, we develop a structure-preserving sub-network to extract the graph-level features. Subsequently, we conduct sample partition over the resultant structure-oriented graph-level embeddings to obtain the original characteristics of each cluster. After that, we upload the multi-source learned prototypes, which are difficult to reconstruct yet follow a Gaussian distribution, to the server. On the server, we design a consensus prototype optimizing scheme to conduct Gaussian estimation on the multi-source condensed structure-oriented information, such that the quality of collaborative learning can be boosted. The updated prototypes are then transferred to each client and combined with the structure-oriented representations to facilitate a robust clustering distribution, thereby enhancing the graph-level clustering performance of local models. The main contributions of this paper are summarized as follows:

- **New research task:** We point out a new research task, i.e., federated graph-level clustering, which makes the first attempt to achieve cross-domain federated graph learning under unsupervised circumstances.
- **Novel FGL framework:** We propose an unsupervised FGL framework, termed Federated Graph-level Clustering Network (FedGCN). The consensus prototype optimizing scheme can enhance the quality of multiple-score information negotiation to promote each local model generating a robust clustering distribution.
- **Better clustering results:** Extensive experiments have been conducted on fifteen cross-dataset and cross-domain non-IID graph datasets without annotations, demonstrating the effectiveness and superiority of the proposed FedGCN against its competitors.

## Related Work

### Federated Graph Learning

Federated learning (FL) is a widely adopted distributed learning paradigm that enables multiple clients to collaboratively train a machine learning model without sharing private data (Wang et al. 2021a,b, 2023, 2024b,c,a). Inspired by its success in various applications, researchers in graph machine learning have increasingly explored its potential for graph data analysis. For example, GCFL (Xie et al. 2021) proposes a cross-dataset and cross-domain federated graph-level classification framework that leverages shared structural knowledge to construct connections between graph data from multiple clients, which effectively improves the performance of each local model. Moreover, in situations with diverse data distributions and disproportionate datasets, PGFL (Gauthier et al. 2023) improves the client’s classification performance by calculating the similarities between different local models. Similarly, FGGP (Wan, Huang, and Ye 2024) decouples the global model into two personal levels, with the former capturing domain information through prototypes and the latter introducing global knowledge to obtain

more powerful prototypes. Previous studies have demonstrated that FGL techniques can improve the performance of graph learning tasks across clients, such as model parameter aggregation strategy (Liu et al. 2024), personalized federated learning method (Chen et al. 2022), and feature-structure decoupling scheme (Tan et al. 2023). However, in real-world scenarios, the lack of data annotations hinders the quality of multi-source information negotiation, resulting in sub-optimal feedback to clients. In contrast, we make the first attempt to analyze cross-dataset and cross-domain unlabeled graph data in a collaborative learning manner. In our federated learning framework, the server receives structure-oriented compressed signals from the local models and adopts Gaussian estimation to fit the consensus representations that are then fed back to the clients.

### Deep Graph-Level Clustering

Recently, significant progress has been made in node-level clustering tasks (Tu et al. 2024c, 2022; Liu et al. 2022b; Liang et al. 2023; Tu et al. 2024d; Guan et al. 2024b; Liang et al. 2024; Gong et al. 2024, 2022). However, the clustering problem across multiple graphs remains underexplored, despite its significant importance in numerous real-world applications. Consequently, graph-level clustering has attracted increasing attention from graph machine-learning researchers. GLCC (Ju et al. 2023) is the first model proposed to solve the multi-graph clustering task, which constructs an adaptive affinity graph to facilitate instance-cluster comparison learning. Subsequently, to address the issue of clustering collapse caused by insufficient discriminability in graph-level representations, UDGC (Hu et al. 2023) evenly distributes instances across different clusters and projects these clusters onto a unit hypersphere, achieving a more uniform distribution at the clustering level. Additionally, researchers have observed that spectral clustering based on graph kernels remains an intuitive and effective approach for graph-level clustering (Togninalli et al. 2019). However, most existing graph kernel methods rely on manual design, which limits their generalization ability across various types of graphs, resulting in unsatisfactory clustering results. To overcome this limitation, DGLC (Cai et al. 2024) introduces a pseudo-label-guided mutual information maximization network, enabling more effective differentiation of graph-level clustering representations. Although existing methods have achieved great success in graph-level clustering, these methods require processing multiple datasets from diverse sources within a single model, which inevitably increases the risk of privacy leakage. Differently, we integrate federated learning with graph-level clustering tasks to address these concerns. To ensure privacy protection, we share multiple prototypes that are inherently difficult to reconstruct, facilitating the establishment of structural correlations among non-IID graph data.

## The Proposed Methodology

### Preliminaries

**Gaussian Mixture Model** It combines multiple Gaussian distributions and estimates their parameters by maximizing

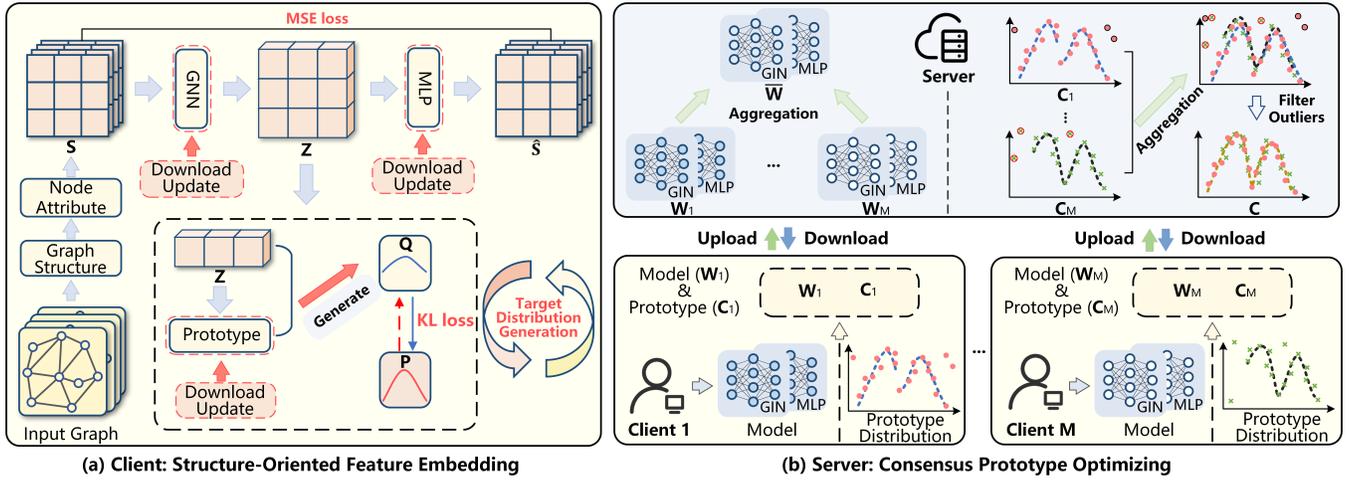


Figure 1: The overall architecture of FedGCN. The structure-oriented feature embedding and consensus prototype optimizing are integrated into a unified alternating optimization FGL framework. Specifically, the structure-oriented model is constructed and uploads the learned prototypes that are hard to reconstruct from the raw data to the server. The consensus prototype optimizing scheme is designed to obtain consensus prototypes from multiple structure-oriented signals through Gaussian estimation that are subsequently transferred to each client to promote the local model for better clustering.

the likelihood function to achieve the best-fitting model. The probability density  $p(x)$  of a data point  $x$  in the mixture model is the weighted sum of the probability densities of all  $k$  components:

$$p(x) = \sum_{k=1}^k \pi_k \mathcal{N}(x|\mu_k, \Sigma_k), \quad (1)$$

where  $\mathcal{N}(x|\mu_k, \Sigma_k)$  denotes standard Gaussian distribution,  $\Sigma_k$  is the covariance matrix,  $\pi_k$  is the mixing coefficient, and  $\mu_k$  is the mean vector of the  $k$ -th component.

**Structure-Oriented Feature Pre-Processing** Given a set of  $I$  undirected graphs  $\mathcal{G} = \{G_1, G_2, \dots, G_I\}$ , where the  $i$ -th graph is  $G_i = (V_i, E_i)$ . Its node attribute matrix is  $\mathbf{X} \in \mathbb{R}^{N \times d_a}$  and the original adjacency matrix is  $\mathbf{A} \in \mathbb{R}^{N \times N}$ , where the  $N$  refers to the number of nodes in a graph and  $d_a$  refers to attribute dimension. The corresponding degree matrix is denoted as  $\mathbf{D} \in \mathbb{R}^{N \times N}$ .

To extract the graph structure features, we apply the FedStar (Tan et al. 2023) method, which leverages degree distribution and random walk encoding techniques. Specifically, we first calculate the degree distribution matrix  $\mathbf{S}_d \in \mathbb{R}^{N \times d_d}$ , where  $d_d$  is the feature dimension of  $\mathbf{S}_d$ :

$$\mathbf{S}_{d[i,j]} = \begin{cases} 1, & \text{if } j = \min(\max(\mathbf{D}_{ii}, 1), N_{dd}) \\ 0, & \text{otherwise} \end{cases}. \quad (2)$$

Next, we obtain the encoded feature matrix  $\mathbf{S}_r \in \mathbb{R}^{N \times d_r}$  using the random walk algorithm and fuse these features via concatenation operation:

$$\mathbf{S}_r = \text{CONCAT}(\mathbf{s}_{r1}, \mathbf{s}_{r2}, \dots, \mathbf{s}_{rN}), \quad (3)$$

where  $\mathbf{s}_r \in \mathbb{R}^{d_r}$  represents the  $n$ -th sample generated by the random walk algorithm.  $d_r$  is the feature dimension of

$\mathbf{S}_r$ . Finally,  $\mathbf{S}_d$ ,  $\mathbf{S}_r$ , and  $\mathbf{X}$  are fused through concatenation operation to obtain the node-level structure-oriented feature matrix  $\mathbf{F} \in \mathbb{R}^{N \times d}$ :

$$\mathbf{F} = \begin{cases} \text{CONCAT}(\mathbf{S}_d, \mathbf{S}_r), & \text{if } \mathbf{X} = \text{None} \\ \text{CONCAT}(\mathbf{S}_d, \mathbf{S}_r, \mathbf{X}), & \text{otherwise} \end{cases}, \quad (4)$$

where  $d$  is the input dimension of the node. It is worth noting that the attribute feature matrix  $\mathbf{X}$  is not present in all datasets. In the absence of attribute features, we use only graph structure features.

### Federated Graph-Level Clustering Network

Fig. 1 shows the architecture of the proposed Federated Graph-Level Clustering Network (FedGCN). The core idea is to collect topology-oriented features from multiple clients and generate global consensus prototypes, which are subsequently fed back to each client to facilitate clustering-friendly graph embeddings.

**Structure-Oriented Feature Embedding** Due to the lack of clear label guidance, the discrepancy in the data distribution during prototype aggregation is amplified when non-IID graph data from multiple clients is processed at the central server. This further limits the performance of local models, which in turn hinders the sharing of multi-source information across clients. Inspired by FedStar (Tan et al. 2023), we develop a GNN-based model that focuses on extracting structure features, as illustrated in Fig. 1(a). Firstly, we employ a GNN model to obtain the structure-oriented embedding matrix for each node:

$$\mathbf{h}_v^{(l)} = \phi\left(\sigma\left(\mathbf{h}_v^{(l-1)}, \omega(\{\mathbf{h}_u^{(l-1)} : u \in \Omega(v)\})\right)\right), \quad (5)$$

where  $\mathbf{h}_v^{(l)} \in \mathbb{R}^{N \times d'}$  denotes the structure-oriented embedding of  $v$ -th node in the  $l$ -th layer,  $d'$  is sample embedding dimension.  $\Omega(v)$  denotes the aggregation operation of

---

**Algorithm 1: Federated Graph-Level Clustering Network**


---

**Input:** Graphs  $\mathcal{G}$ ; degree distribution matrix  $\mathbf{S}_d$ ; random walk feature matrix  $\mathbf{S}_r$ ; attribute matrix  $\mathbf{X}$ ; server epoch  $E_{global}$ ; client epoch  $E_{local}$ ; client number  $M$ .  
**Output:** Clustering results  $\mathbf{O}$  at clients.

- 1: Initialize  $\mathbf{C}^{(0)}$  and  $\mathbf{W}^{(0)}$  at the server.
- 2: **for**  $i = 1$  to  $E_{global}$  **do**
- 3:   **for**  $m$  in  $M$  **do**
- 4:     Generate  $\mathbf{F}$  with  $\mathbf{S}_d$ ,  $\mathbf{S}_r$  and  $\mathbf{X}$  by Eq. (4).
- 5:     Obtain  $\mathbf{C}_m^{(0)}$  and  $\mathbf{W}_m^{(0)}$ .
- 6:     **for**  $j = 1$  to  $E_{local}$  **do**
- 7:       Obtain  $\mathbf{H}$  from  $\mathbf{F}$  by Eq. (5).
- 8:       Obtain  $\mathbf{Z}$  from  $\mathbf{H}$  by Eq. (6).
- 9:       Reconstruct  $\hat{\mathbf{F}}$  from  $\mathbf{H}$  by Eq. (7).
- 10:       Calculate  $\mathbf{Q}$  and  $\mathbf{P}$  from  $\mathbf{Z}$ .
- 11:       Calculate the  $\mathcal{L}_{KL}$  and  $\mathcal{L}_{rec}$ , respectively.
- 12:       Update FedGCN by minimizing Eq.(15).
- 13:       Stack  $\mathbf{C}_m^{(j)}$  to  $\mathbf{C}_m$ .
- 14:     **end for**
- 15:   **end for**
- 16:   Obtain  $\mathbf{U}$  from  $\mathbf{C}$  by Eq. (10).
- 17:   Collect the  $\mathbf{U}$  and  $\mathbf{W}$  of all clients.
- 18:   Calculate  $\mathbf{C}$  using Improved GMM by Eqs. (10)-(14).
- 19:   Aggregate the model weights  $\overline{\mathbf{W}}$ .
- 20: **end for**
- 21: Obtain clustering results  $\mathbf{O}$  at clients by  $K$ -means.
- 22: **return**  $\mathbf{O}$

---

$v$ -th node's neighbors. It is worth noting that  $\mathbf{h}_v^{(0)}$  is initialized with  $v$ -th node structure-oriented features  $\mathbf{f}_v$ .  $\omega(\cdot)$ ,  $\sigma(\cdot)$ , and  $\phi(\cdot)$  denote the aggregation function, updating function, and normalization function, respectively. Subsequently, We concatenate the sums of node-level representations across all layers and then feed it into the Multi-Layer Perceptron (MLP) network to obtain the graph-level embedding matrix  $\mathbf{Z} \in \mathbb{R}^{N \times d'}$ :

$$\mathbf{Z} = \text{MLP}[\text{CONCAT}(\sum_{v \in G} \mathbf{h}_v^{(1)}, \dots, \sum_{v \in G} \mathbf{h}_v^{(L)})]. \quad (6)$$

Subsequently, we reconstruct the structure-oriented feature matrix, denoted as  $\hat{\mathbf{F}} \in \mathbb{R}^{N \times d}$ . Finally, we calculate the mean squared error between  $\mathbf{F}$  and  $\hat{\mathbf{F}}$  to ease the learning process of the local model:

$$\hat{\mathbf{F}} = \text{MLP}(\mathbf{H}^{(L)}), \quad (7)$$

$$\mathcal{L}_{MSE} = \frac{1}{2N} \|\mathbf{F} - \hat{\mathbf{F}}\|_2. \quad (8)$$

In summary, the proposed structure-oriented feature embedding model offers the following key advantages: 1) it enables the model to upload local graph data features to the server without sharing private data while preserving the clustering characteristics of each cluster; and 2) it ensures domain invariance across graph datasets, effectively capturing common patterns within the non-IID graph data without annotations.

**Consensus Prototype Optimizing** As mentioned in the literature (Wan, Huang, and Ye 2024), prototypes typically preserve the clustering characteristics of graph data from different clients. Building on this idea, we aim to collect graph prototypes and well-trained model parameters from multiple clients and subsequently share these signals at the server. To this end, we develop a consensus prototype optimizing scheme within the server aggregation pipeline of FedGCN, as illustrated in Figure 1(b). Specifically, in the  $m$ -th client, the prototypes  $\mathbf{C}_m \in \mathbb{R}^{N \times d'}$  are obtained by using the  $\mathbf{Z}$ . Subsequently, the results and the model parameters  $\mathbf{W}_m$  are then uploaded to the server. Next, at the server, prototype sharing and model aggregation are performed to obtain consensus results that capture shared useful information among clients. Later, we calculate the stacked prototype matrix  $\mathbf{U} \in \mathbb{R}^{N \times M d'}$  from  $[\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_M]$ :

$$\mathbf{U} = \text{CONCAT}(\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_M). \quad (9)$$

Next, according to multiple prototypes  $\mathbf{U}$  from clients, we adopt the Gaussian estimation to calculate and update the consensus prototype representations. However, in practical applications, graph data often exhibits characterized by non-strict Gaussian distributions and outliers (Peng et al. 2024). To alleviate these effects, we improve the Expectation-Maximization (EM) algorithm of the Gaussian Mixture Model (GMM). In the  $E$ -step, we incorporate Kernel Density Estimation (KDE) (Zandieh et al. 2023) to more accurately approximate the true data distribution and effectively filter out outliers. Specifically, the probability density of sample  $\mathbf{U}$  is calculated via the KDE function  $f(\cdot)$ :

$$f(\mathbf{u}_i) = \frac{\sum_{j=1}^T \exp(-\frac{1}{2}(\mathbf{u}_i - \mathbf{u}_j)^\top \Sigma_k^{-1}(\mathbf{u}_i - \mathbf{u}_j))}{bT(2\pi)^{d'/2} \sum_k \pi_k}, \quad (10)$$

where  $\exp(\cdot)$  is exponential function,  $\mathbf{u}_i$  represents the  $i$ -th component of  $\mathbf{U}$ ,  $T$  represents the number of samples,  $b$  refers to the bandwidth (i.e., smoothing parameter), and  $\sum_k$  is the variance of the  $k$ -th component. Subsequently, we update the responsibility  $\gamma_{ik}$ :

$$\gamma_{ik} = \frac{\pi_k f_k(\mathbf{u}_i)}{\sum_{j=1}^K \pi_j f_j(\mathbf{u}_j)}, \quad (11)$$

where  $K$  denotes the number of components,  $f_k(\cdot)$  and  $f_j(\cdot)$  represent the KDE functions used for the  $k$ -th and  $j$ -th components, respectively. In the  $M$ -step, we design a learnable parameter  $\alpha_{ik}$  to adjust the smoothing degree:

$$\alpha_{ik} = \exp(-\beta \cdot (\frac{\gamma_{ik}}{\sum_{j=1}^K \gamma_{ij}})), \quad (12)$$

$$\pi_k^{\text{new}} = \frac{\sum_{i=1}^T \alpha_{ik} \gamma_{ik}}{\sum_{i=1}^T \alpha_{ik}}, \mu_k^{\text{new}} = \frac{\sum_{i=1}^T \alpha_{ik} \gamma_{ik} \mathbf{u}_i}{\sum_{i=1}^T \alpha_{ik} \gamma_{ik}}, \quad (13)$$

$$\Sigma_k^{\text{new}} = \frac{\sum_{i=1}^T \alpha_{ik} \gamma_{ik} (\mathbf{u}_i - \mu_k^{\text{new}})(\mathbf{u}_i - \mu_k^{\text{new}})^\top}{\sum_{i=1}^T \alpha_{ik} \gamma_{ik}}, \quad (14)$$

where  $\beta$  is a scaling factor. Finally, the network is updated by alternately performing  $E$ -step and  $M$ -step. Upon convergence, the updated mean  $\mu$  corresponds to the consensus

prototypes  $\mathbf{C} \in \mathbb{R}^{N \times d'}$ . The server then transmits these consensus prototypes  $\mathbf{C}$  back to each client.

The model aggregation process follows a similar approach to FedAvg (McMahan et al. 2017). The model parameters  $\{\mathbf{W}_m\}_{m=1}^M$  trained on the clients are uploaded to the server. The server aggregates the local models using  $\overline{\mathbf{W}} = \sum_{m=1}^M \frac{R_m}{R_{tot}} \mathbf{W}_m$  to compute the consensus model parameters  $\overline{\mathbf{W}}$ , which are then distributed back to the clients. Here,  $R_m$  represents the number of graphs in  $m$ -th client, and  $R_{tot}$  denotes the number of graphs across all clients.

The merits of the proposed consensus prototype optimization scheme can be summarized as: 1) without sharing privacy data, the model uploads local graph data features that preserve the clustering characteristics of each cluster to the server; and 2) in the absence of label guidance, the network effectively explores the latent structural patterns within each client’s non-IID graph data, decreasing the risk of global sharing failure caused by variations in attribute information.

**Client-Server Collaborative Learning** After information aggregation at the server, each client receives the updated  $\mathbf{C}$  and  $\overline{\mathbf{W}}$  from the global model. Subsequently,  $\mathbf{C}$  serves as the cluster centers for the next round of local model training, and  $\overline{\mathbf{W}}$  is used as the initialization for the model parameters. In this way, the quality of the learned local representations could be further improved. In the collaborative learning phase, the clustering loss function is defined by minimizing the Kullback-Leibler (KL) divergence between  $\mathbf{P}$  and  $\mathbf{Q}$  (Tu et al. 2021):

$$\mathcal{L}_{KL} = \text{KL}(\mathbf{P}||\mathbf{Q}) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}, \quad (15)$$

where the soft assignment distribution  $\mathbf{Q}$  of  $\mathbf{Z}$  is calculated using the Student’s  $t$ -distribution (Guan et al. 2024c). The target distribution  $\mathbf{P}$  represents that the embedding of the  $i$ -th sample belongs to the  $j$ -th prototype.

**Loss Function and Optimization** The overall learning objective consists of two main components: the KL divergence loss  $\mathcal{L}_{KL}$  and the reconstruction loss  $\mathcal{L}_{MSE}$ :

$$\mathcal{L} = \mathcal{L}_{KL} + \lambda \mathcal{L}_{MSE}, \quad (16)$$

where  $\lambda$  is a predefined hyperparameter that balances the importance of reconstruction and clustering. Algorithm 1 provides a detailed description of the FedGCN learning process.

## Experiments

### Experiment Setup

**Benchmark Datasets** To verify the effectiveness of our method, we employ 15 benchmark graph datasets across different domains, including Small Molecules (e.g., MUTAG, BZR, COX2, DHFR, PTC\_MR, AIDS, BZR\_MD), Bioinformatics (e.g., DD, PROTEINS), Synthetic (e.g., SYNTHETIC), Social Networks (e.g., COLLAB, IMDB-MULTI), and Computer Vision (e.g., Letter-high, Letter-low, Letter-med) (Morris et al. 2020). Table 1 summarizes the detailed information of the above datasets. In our setup, five types of non-IID settings are created: 1) 2 clusters within

Datasets	Domain	Classes	Graphs	A.Nodes	A.Edges
MUTAG	SM	2	188	17.93	19.79
BZR			405	35.75	38.36
COX2			467	41.22	43.45
DHFR			756	42.43	44.54
PTC_MR			344	14.29	14.69
AIDS			2000	15.69	16.20
BZR_MD			306	21.30	225.06
DD	BIO	2	1178	284.32	715.66
PROTEINS			1113	39.06	72.82
SYNTHETIC	SY	2	300	100.00	196.00
COLLAB	SN	3	5000	74.49	2457.78
IMDB-MULTI			1500	13.00	65.94
Letter-high	CV	15	2250	4.67	4.50
Letter-low			2250	4.68	3.13
Letter-med			2250	4.67	3.21

Table 1: Description of all used datasets.

the same domain (SM); 2) 3 clusters within the same domain (SN); 3) 15 clusters within the same domain (CV); 4) 2 clusters across two domains (SM-BIO); and 5) 2 clusters across three domains (SM-BIO-SY).

**Baseline Methods** We compare FedGCN with three classical federated learning methods, including FedAvg (McMahan et al. 2017), FedProx (Li et al. 2020), and FedPer (Ariavzhagan et al. 2019). In addition, we also involve three other state-of-the-art federated graph learning methods, including FedSage (Zhang et al. 2021), GCFL (Xie et al. 2021), and FedStar (Tan et al. 2023).

**Implementation Details** To ensure experimental fairness, the results of FedGCN and all compared methods are obtained under the same device and configuration settings. We employ a three-layer GIN (Xu et al. 2018) to obtain the graph-level structure-oriented embedding, with the hidden layer dimension set to 64 and a batch size of 128 for each local model. During model optimization, we use the Adam optimizer (Kingma and Ba 2014) with a learning rate of 1e-3. The client interacts with the server 10 times, performing 10 training epochs during each interaction. All methods are implemented using PyTorch, and experiments are conducted on a single NVIDIA GeForce RTX 4090 GPU.

**Evaluation Metrics** We employ four widely-used clustering metrics to evaluate the performance of all compared methods: Accuracy (ACC) (Li and Ding 2006; Tu et al. 2024b; Sun et al. 2021; Liu et al. 2022a), Normalized Mutual Information (NMI) (Strehl and Ghosh 2002; Liu et al. 2023; Guan et al. 2024a), Adjusted Rand Index (ARI) (Hubert and Arabie 1985; Liu et al. 2021; Tu et al. 2024a), and F1 Score (F1) (Zhou et al. 2019; Liang et al. 2024). These metrics test clustering results from different perspectives, with higher values indicating better clustering performance.

### Experimental Results

**Comparison in Unsupervised Learning Settings** To validate the effectiveness of the proposed FedGCN, which is the first federated graph-level clustering framework, we

Classes	Domain	Metric	FedSage	GCFL	FedStar	FedGCN
			NeurIPS21	NeurIPS21	AAAI23	Ours
2	SM (7)	ACC	55.6±1.4	61.1±1.8	58.9±2.4	<b>75.9±0.8</b>
		NMI	12.2±1.3	8.7±2.4	12.0±1.2	<b>24.9±3.0</b>
		ARI	7.6±0.6	9.4±2.4	0.1±0.8	<b>31.1±3.4</b>
		F1	50.2±1.0	43.3±1.6	49.7±2.8	<b>67.1±1.5</b>
	SM-BIO (9)	ACC	57.4±2.2	60.1±1.8	59.5±1.6	<b>69.2±0.6</b>
		NMI	5.2±2.1	4.7±2.4	5.3±1.6	<b>14.0±2.7</b>
		ARI	4.2±2.7	3.2±2.3	3.8±2.0	<b>17.5±3.1</b>
		F1	49.9±0.5	47.3±1.5	51.7±2.2	<b>59.1±0.9</b>
	SM-BIO-SY (10)	ACC	57.6±1.9	59.1±2.0	57.9±2.6	<b>68.6±1.3</b>
		NMI	20.6±1.9	14.4±2.2	15.7±2.4	13.5±2.1
		ARI	17.6±2.4	13.7±2.8	16.1±3.0	17.2±3.6
		F1	49.4±1.7	52.3±1.9	52.3±2.2	<b>59.4±3.8</b>
3	SN (2)	ACC	53.3±1.9	52.1±2.3	51.7±2.7	<b>66.6±2.3</b>
		NMI	14.8±1.4	12.5±2.3	13.7±2.8	<b>30.4±6.6</b>
		ARI	11.6±2.8	13.2±2.3	12.4±1.9	<b>34.1±5.3</b>
		F1	49.3±2.0	52.3±1.6	50.7±2.3	50.7±2.4
15	CV (3)	ACC	10.1±1.4	13.7±2.0	12.4±2.7	<b>34.6±2.8</b>
		NMI	30.5±1.7	17.7±2.4	22.4±2.5	<b>34.2±1.4</b>
		ARI	13.6±1.8	14.3±2.7	15.3±2.1	<b>19.3±1.8</b>
		F1	10.4±1.7	13.2±1.4	11.6±1.9	<b>31.6±3.1</b>

Table 2: Performance comparison across different federated graph-level clustering methods. Notably, all compared methods are evaluated under unsupervised settings on cross-dataset or cross-domain datasets to ensure a fair comparison.

adopt three representative supervised FGL methods and extend them into unsupervised versions to ensure a fair comparison. To minimize randomness, each experiment is repeated five times, and the mean and standard deviation of the four clustering metrics are presented. The experimental results are presented in Table 2. As seen, we report the results of all methods on federated graph-level clustering across five non-IID settings, including cross-dataset datasets (i.e., SM, SN, CV), across two domains (i.e., SM-BIO), and across three domains (i.e., SM-BIO-SY). We can observe that FedGCN outperforms compared methods in most cases. The reasons behind this are two-fold: 1) although methods such as FedSage, GCFL, and FedStar are specifically designed to enhance connections between local models, these approaches are tailored for supervised settings. As a result, their effectiveness is limited when dealing with non-IID graph data without annotations; and 2) on one hand, the proposed method is uniquely designed for unsupervised non-IID graph scenarios. On the other hand, the prototype optimizing scheme establishes useful connections between non-IID graph data across multiple clients while ensuring privacy, thus generating clustering-friendly prototypes to guide each local model in producing robust clustering distribution.

**Comparison with Supervised FGL Methods** To further validate the superiority of FedGCN, we compare it with three supervised FGL methods: FedSage, GCFL, and FedStar. Each of these methods is provided with a limited amount of labeled data. Specifically, for two-class and three-class classification tasks, five true labels are assigned to each class, while for the fifteen-class classification task, one true label is assigned to each class. The experimental results pre-

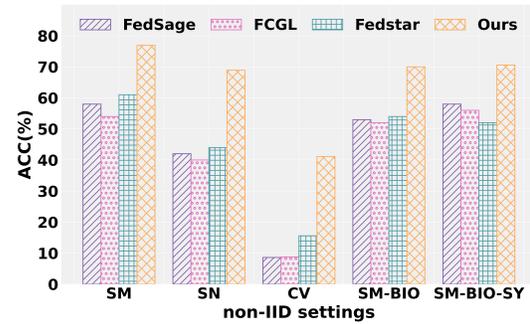


Figure 2: Performance comparison between three advanced federated graph-level classification methods and FedGCN on five types of non-IID settings.

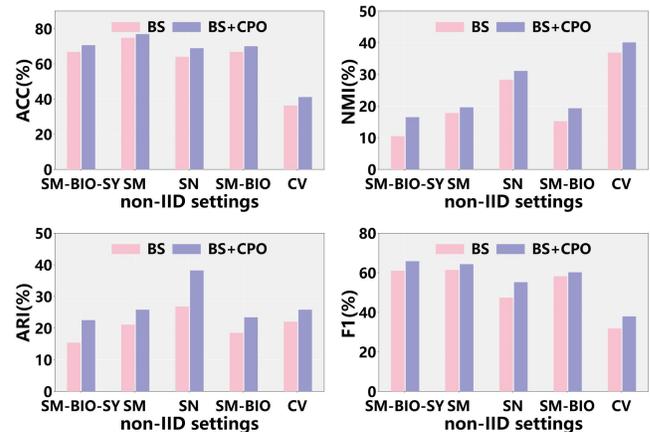


Figure 3: Ablation study for the consensus prototype optimizing scheme in unsupervised FGL scenarios. “BS” denotes the client model of FedGCN and “BS + CPO” denotes the baseline method with the consensus prototype optimizing scheme.

sented in Fig. 2 demonstrate that FedGCN consistently outperforms its competitors in clustering performance across both cross-dataset and cross-domain scenarios, further highlighting the effectiveness of the proposed method compared to the supervised approaches.

### Effect of the Consensus Prototype Optimizing Scheme

In this part, we evaluate the effectiveness of the proposed consensus prototype optimizing scheme. In our setup, “BS” denotes the client model of FedGCN, and “BS+CPO” denotes the baseline method with the consensus prototype optimizing scheme. As shown in Fig. 3, we can conclude that the “BS + CPO” method consistently outperforms the “BS” method in terms of four clustering metrics across all datasets. These improvements could be attributed to the following points. Firstly, the consensus prototype optimizing scheme effectively establishes structure-oriented correlations between clients at the server, enabling the underlying connectivity information to be transferred to each local

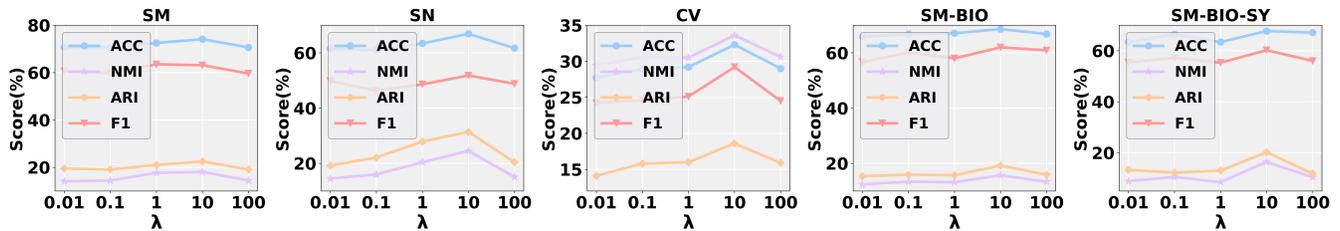


Figure 4: The sensitivity of FedGCN with the variation of a hyper-parameter  $\lambda$  on five non-IID settings.

Domains	Model	PS	ACC	NMI	ARI	F1	$\Delta$
SM (7)	FedProx	-	73.3±0.8	18.4±3.0	21.7±3.3	63.0±2.2	
	FedProx	✓	73.9±0.8	21.0±2.6	25.3±3.0	65.1±1.9	↑2.2
	FedPer	-	72.3±0.3	17.6±1.2	21.7±1.6	62.9±0.8	
	FedPer	✓	73.5±0.3	19.3±1.8	25.6±2.7	65.1±2.0	↑2.3
	FedAvg	-	71.6±0.7	16.3±2.8	20.4±1.5	62.1±1.0	
	FedAvg	✓	<b>75.9±0.8</b>	<b>24.9±3.0</b>	<b>31.1±3.4</b>	<b>67.1±1.5</b>	↑7.2
SM-BIO (9)	FedProx	-	68.0±0.6	11.5±1.0	15.1±1.0	59.8±2.2	
	FedProx	✓	68.8±0.5	14.4±0.7	17.9±0.6	60.0±0.9	↑1.7
	FedPer	-	68.4±0.4	10.5±1.3	13.5±0.5	56.7±1.0	
	FedPer	✓	68.7±0.1	11.7±1.9	15.4±1.9	58.9±3.0	↑1.4
	FedAvg	-	66.7±0.2	11.1±1.3	13.8±1.1	57.4±1.5	
	FedAvg	✓	<b>69.5±0.7</b>	<b>14.9±2.2</b>	<b>18.9±2.7</b>	<b>61.3±1.3</b>	↑3.9

Table 3: Ablation studies for the prototype-sharing component. “✓” indicates the compared method using prototype-sharing (i.e., PS) component. “↑” denotes the average performance improvement in terms of four metrics.

model for better clustering performance. Secondly, the improved Gaussian estimation approach mitigates the impact of non-strict Gaussian distributions and outliers, enabling each local model to generate a more reliable target distribution guided by the updated prototype.

**Effect of the Prototype-Sharing Component** In this subsection, we conduct ablation studies to investigate the effectiveness of the prototype-sharing component. In our setup, local models are employed to generate prototypes, which are then shared using federated learning strategies from popular methods, including FedProx, FedPer, and FedAvg. From the results in Table 3, we can see that regardless of the federated learning strategy employed, integrating the proposed prototype-sharing component enhances clustering performance on both cross-dataset and cross-domain datasets. For example, in the SM setting, the method incorporating the prototype-sharing component achieves an average improvement of up to 7.2%, while in the SM-BIO setting, it achieves an average improvement of up to 3.9%. Moreover, FedAvg demonstrates much better clustering performance compared to FedProx and FedPer across multiple datasets, indicating the model-aggregation-based federated learning strategy contributes more to the model performance. These findings suggest that the integration of the prototype-sharing and model-aggregation components in FedGCN substantially enhances the generalization ability of graph-level federated clustering frameworks.

**Analysis of Hyper-Parameter  $\lambda$**  In Eq. (15), FedGCN introduces a hyper-parameter  $\lambda$  to balance two losses. We conduct experiments on multiple datasets to analyze the effect of this hyper-parameter by varying it from 0.01 to 100. From the results in Fig. 4, we can observe that 1) the hyper-parameter  $\lambda$  indeed plays a crucial role in FedGCN, suggesting that searching  $\lambda$  from a reasonable hyper-parameter region could benefit clustering performance; 2) the model performance is relatively stable in a wide range of  $\lambda$ , indicating that an appropriate  $\lambda$  value is required to balance two learning processes; 3) when  $\lambda$  is set to 10, FedGCN tends to achieve better clustering performance across all datasets.

## Conclusion

In this paper, we point out a new research task, i.e., federated graph-level clustering, which is the first attempt to achieve cross-domain federated graph learning (FGL) under unsupervised circumstances. To handle this task, we propose a novel unsupervised FGL framework termed FedGCN. In our method, the proposed structure-oriented feature embedding model and consensus prototype optimizing scheme effectively enhance the quality of multi-source information negotiation at the server, even in the absence of data annotation guidance, leading to improved feedback to clients for better clustering. Extensive experiments on multiple non-IID graph datasets demonstrate that FedGCN effectively analyzes non-IID graphs in unsupervised settings while ensuring robust data privacy protection. In future work, we aim to further improve FedGCN to address more challenging graph learning tasks, such as incomplete federated graph clustering and federated node-level clustering.

## Acknowledgments

This work is supported by the Key Research and Development Program of Hainan Province (Grant No. ZDYF2024GXJS014, ZDYF2023GXJS163), the National Natural Science Foundation of China (Grant No. 62162022, 62162024), the Natural Science Foundation of Hainan University (Grant No. XJ2400009401), the Youth Foundation Project of Hainan Natural Science Foundation (Grant No. 624QN279), and the Innovative research project for Graduate students in Hainan Province (Grant No. Qhyb2023-28). Corresponding authors: Wenxuan Tu and Jieren Cheng.

## References

- Arivazhagan, M. G.; Aggarwal, V.; Singh, A. K.; and Choudhary, S. 2019. Federated learning with personalization layers. In *arXiv preprint arXiv:1912.00818*.
- Cai, J.; Han, Y.; Guo, W.; and Fan, J. 2024. Deep graph-level clustering using pseudo-label-guided mutual information maximization network. *Neural Computing and Applications*, 36(16): 9551–9566.
- Chen, F.; Long, G.; Wu, Z.; Zhou, T.; and Jiang, J. 2022. Personalized federated learning with graph. In *arXiv preprint arXiv:2203.00829*.
- Chen, X.; Chen, S.; Yao, J.; Zheng, H.; Zhang, Y.; and Tsang, I. W. 2020. Learning on attribute-missing graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(2): 740–757.
- Cheung, T.-H.; Dai, W.; and Li, S. 2021. Fedsgc: Federated simple graph convolution for node classification. In *Proceedings of International Joint Conference on Artificial Intelligence Workshops*.
- Fredrikson, M.; Jha, S.; and Ristenpart, T. 2015. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the ACM SIGSAC conference on computer and communications security*, 1322–1333.
- Gauthier, F.; Gogineni, V. C.; Werner, S.; Huang, Y.-F.; and Kuh, A. 2023. Personalized graph federated learning with differential privacy. *IEEE Transactions on Signal and Information Processing over Networks*, 9: 736–749.
- Gong, L.; Tu, W.; Zhou, S.; Zhao, L.; Liu, Z.; and Liu, X. 2024. Deep Fusion Clustering Network With Reliable Structure Preservation. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, 35(6): 7792–7803.
- Gong, L.; Zhou, S.; Tu, W.; and Liu, X. 2022. Attributed Graph Clustering with Dual Redundancy Reduction. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 3015–3021.
- Guan, R.; Li, Z.; Li, X.; and Tang, C. 2024a. Pixel-superpixel contrastive learning and pseudo-label correction for hyperspectral image clustering. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 6795–6799.
- Guan, R.; Li, Z.; Tu, W.; Wang, J.; Liu, Y.; Li, X.; Tang, C.; and Feng, R. 2024b. Contrastive Multiview Subspace Clustering of Hyperspectral Images Based on Graph Convolutional Networks. *IEEE Transactions on Geoscience and Remote Sensing*, 62: 1–14.
- Guan, R.; Tu, W.; Li, Z.; Yu, H.; Hu, D.; Chen, Y.; Tang, C.; Yuan, Q.; and Liu, X. 2024c. Spatial-Spectral Graph Contrastive Clustering with Hard Sample Mining for Hyperspectral Images. *IEEE Transactions on Geoscience and Remote Sensing*, 1–16.
- Hu, C.; Liu, B.; Zhang, X.; Wang, Z.; Lin, C.; and Luo, L. 2022. A federated multi-server knowledge graph embedding framework for link prediction. In *Proceedings of IEEE International Conference on Tools with Artificial Intelligence*, 366–371.
- Hu, M.; Chen, C.; Liu, W.; Zhang, X.; Liao, X.; and Zheng, X. 2023. Learning Uniform Clusters on Hypersphere for Deep Graph-level Clustering. In *arXiv preprint arXiv:2311.13953*.
- Huang, W.; Wan, G.; Ye, M.; and Du, B. 2024. Federated graph semantic and structural learning. In *arXiv preprint arXiv:2406.18937*.
- Hubert, L.; and Arabie, P. 1985. Comparing partitions. *Journal of classification*, 2: 193–218.
- Ju, W.; Gu, Y.; Chen, B.; Sun, G.; Qin, Y.; Liu, X.; Luo, X.; and Zhang, M. 2023. Glcc: A general framework for graph-level clustering. In *Proceedings of the AAAI conference on artificial intelligence*, 4391–4399.
- Kingma, D.; and Ba, J. 2014. Adam: A Method for Stochastic Optimization. In *arXiv preprint arXiv:1412.6980*.
- Li, T.; and Ding, C. 2006. The relationships among various nonnegative matrix factorization methods for clustering. In *Proceedings of International Conference on Data Mining*, 362–371.
- Li, T.; Sahu, A. K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; and Smith, V. 2020. Federated optimization in heterogeneous networks. In *Proceedings of Machine learning and systems*, 429–450.
- Liang, K.; Liu, Y.; Zhou, S.; Tu, W.; Wen, Y.; Yang, X.; Dong, X.; and Liu, X. 2023. Knowledge graph contrastive learning based on relation-symmetrical structure. *IEEE Transactions on Knowledge and Data Engineering*, 36(1): 226–238.
- Liang, K.; Meng, L.; Liu, M.; Liu, Y.; Tu, W.; Wang, S.; Zhou, S.; Liu, X.; Sun, F.; and He, K. 2024. A survey of knowledge graph reasoning on graph types: Static, dynamic, and multi-modal. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12): 9456–9478.
- Liu, J.; Liu, X.; Yang, Y.; Guo, X.; Kloft, M.; and He, L. 2021. Multiview subspace clustering via co-training robust data representation. *IEEE Transactions on Neural Networks and Learning Systems*, 33(10): 5177–5189.
- Liu, J.; Liu, X.; Yang, Y.; Liao, Q.; and Xia, Y. 2023. Contrastive multi-view kernel learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(8): 9552–9566.
- Liu, R.; Xing, P.; Deng, Z.; Li, A.; Guan, C.; and Yu, H. 2024. Federated graph neural networks: Overview, techniques, and challenges. *IEEE Transactions on Neural Networks and Learning Systems*, 1–17.
- Liu, S.; Wang, S.; Zhang, P.; Xu, K.; Liu, X.; Zhang, C.; and Gao, F. 2022a. Efficient one-pass multi-view subspace clustering with consensus anchors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 7576–7584.
- Liu, Y.; Tu, W.; Zhou, S.; Liu, X.; Song, L.; Yang, X.; and Zhu, E. 2022b. Deep Graph Clustering via Dual Correlation Reduction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 7603–7611.
- Luo, M.; Chen, F.; Hu, D.; Zhang, Y.; Liang, J.; and Feng, J. 2021. No fear of heterogeneity: Classifier calibration for

- federated learning with non-iid data. In *Proceedings of Advances in Neural Information Processing Systems*, 5972–5984.
- McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of Artificial intelligence and statistics*, 1273–1282.
- Morris, C.; Kriege, N. M.; Bause, F.; Kersting, K.; Mutzel, P.; and Neumann, M. 2020. TUDataset: A collection of benchmark datasets for learning with graphs. In *arXiv preprint arXiv:2007.08663*.
- Peng, J.; Luo, B.; Xu, L.; Yang, J.; Zhang, C.; and Pei, Z. 2024. Blind Image Deblurring via Minimizing Similarity Between Fuzzy Sets on Image Pixels. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(11): 11851–11873.
- Ren, Y.; Chen, X.; Xu, J.; Pu, J.; Huang, Y.; Pu, X.; Zhu, C.; Zhu, X.; Hao, Z.; and He, L. 2024. A novel federated multi-view clustering method for unaligned and incomplete data fusion. *Information Fusion*, 108: 102357.
- Strehl, A.; and Ghosh, J. 2002. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, 3(12): 583–617.
- Sun, M.; Zhang, P.; Wang, S.; Zhou, S.; Tu, W.; Liu, X.; Zhu, E.; and Wang, C. 2021. Scalable multi-view subspace clustering with unified anchors. In *Proceedings of the ACM international conference on multimedia*, 3528–3536.
- Tan, Y.; Liu, Y.; Long, G.; Jiang, J.; Lu, Q.; and Zhang, C. 2023. Federated learning on non-iid graphs via structural knowledge sharing. In *Proceedings of the AAAI conference on artificial intelligence*, 9953–9961.
- Togninalli, M.; Ghisu, E.; Llinares-López, F.; Rieck, B.; and Borgwardt, K. 2019. Wasserstein weisfeiler-lehman graph kernels. In *Proceedings of Advances in Neural Information Processing Systems*.
- Tu, W.; Guan, R.; Zhou, S.; Ma, C.; Peng, X.; Cai, Z.; Liu, Z.; Cheng, J.; and Liu, X. 2024a. Attribute-Missing Graph Clustering Network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 15392–15401.
- Tu, W.; Liao, Q.; Zhou, S.; Peng, X.; Ma, C.; Liu, Z.; Liu, X.; Cai, Z.; and He, K. 2024b. RARE: Robust Masked Graph Autoencoder. *IEEE Transactions on Knowledge and Data Engineering*, 36(10): 5340–5353.
- Tu, W.; Xiao, B.; Liu, X.; Zhou, S.; Cai, Z.; and Cheng, J. 2024c. Revisiting Initializing Then Refining: An Incomplete and Missing Graph Imputation Network. *IEEE Transactions on Neural Networks and Learning Systems*, 1–14.
- Tu, W.; Zhou, S.; Liu, X.; Ge, C.; Cai, Z.; and Liu, Y. 2024d. Hierarchically Contrastive Hard Sample Mining for Graph Self-supervised Pre-training. *IEEE Transactions on Neural Networks and Learning Systems*, 35(11): 16748–16761.
- Tu, W.; Zhou, S.; Liu, X.; Guo, X.; Cai, Z.; Zhu, E.; and Cheng, J. 2021. Deep fusion clustering network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 9978–9987.
- Tu, W.; Zhou, S.; Liu, X.; Liu, Y.; Cai, Z.; Zhu, E.; Zhang, C.; and Cheng, J. 2022. Initializing Then Refining: A Simple Graph Attribute Imputation Network. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 3494–3500.
- Wan, G.; Huang, W.; and Ye, M. 2024. Federated graph learning under domain shift with generalizable prototypes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 15429–15437.
- Wang, H.; Guo, S.; Tang, B.; Li, R.; Yang, Y.; Qu, Z.; and Wang, Y. 2021a. Heterogeneity-aware gradient coding for tolerating and leveraging stragglers. *IEEE Transactions on Computers*, 71(4): 779–794.
- Wang, H.; Jia, Y.; Zhang, M.; Hu, Q.; Ren, H.; Sun, P.; Wen, Y.; and Zhang, T. 2024a. FedDSE: Distribution-aware Sub-model Extraction for Federated Learning over Resource-constrained Devices. In *Proceedings of the ACM on Web Conference*, 2902–2913.
- Wang, H.; Li, Y.; Xu, W.; Li, R.; Zhan, Y.; and Zeng, Z. 2023. Dafkd: Domain-aware federated knowledge distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 20412–20421.
- Wang, H.; Qu, Z.; Guo, S.; Wang, N.; Li, R.; and Zhuang, W. 2021b. LOSP: Overlap synchronization parallel with local compensation for fast distributed training. *IEEE Journal on Selected Areas in Communications*, 39(8): 2541–2557.
- Wang, H.; Xu, H.; Li, Y.; Xu, Y.; Li, R.; and Zhang, T. 2024b. FedCDA: Federated Learning with Cross-rounds Divergence-aware Aggregation. In *Proceedings of The International Conference on Learning Representations*.
- Wang, H.; Zheng, P.; Han, X.; Xu, W.; Li, R.; and Zhang, T. 2024c. FedNLR: Federated Learning with Neuron-wise Learning Rates. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 3069–3080.
- Xie, H.; Ma, J.; Xiong, L.; and Yang, C. 2021. Federated graph classification over non-iid graphs. In *Proceedings of Advances in Neural Information Processing Systems*, 18839–18852.
- Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2018. How powerful are graph neural networks? In *arXiv preprint arXiv:1810.00826*.
- Yan, X.; Wang, Z.; and Jin, Y. 2024. Federated Incomplete Multi-View Clustering with Heterogeneous Graph Neural Networks. In *arXiv preprint arXiv:2406.08524*.
- Zandieh, A.; Han, I.; Daliri, M.; and Karbasi, A. 2023. Kde-former: Accelerating transformers via kernel density estimation. In *Proceedings of International Conference on Machine Learning*, 40605–40623.
- Zhang, K.; Yang, C.; Li, X.; Sun, L.; and Yiu, S. M. 2021. Subgraph federated learning with missing neighbor generation. In *Proceedings of Advances in Neural Information Processing Systems*, 6671–6682.
- Zhou, S.; Liu, X.; Li, M.; Zhu, E.; Liu, L.; Zhang, C.; and Yin, J. 2019. Multiple kernel clustering with neighbor-kernel subspace segmentation. *IEEE Transactions on Neural Networks and Learning Systems*, 31(4): 1351–1362.