FedCCH: Automatic Personalized Graph Federated Learning for Inter-Client and Intra-Client Heterogeneity

Pengfei Jiao[1] , **Zian Zhou**[1] , **Meiting Xue**[1] , **Huijun Tang**[1] , **Zhidong Zhao**[1] and **Huaming Wu**[2,*]

[1]School of Cyberspace, Hangzhou Dianzi University, Hangzhou, China

[2]Center for Applied Mathematics, Tianjin University, Tianjin, China

{pjiao, zza_1998 , munuan, tanghuijune, zhaozd}@hdu.edu.cn, whming@tju.edu.cn

Abstract

Graph federated learning(GFL) is increasingly utilized in domains such as social network analysis and recommendation systems, where non-IID data exist extensively and necessitate a strong emphasis on personalized learning. However, existing methods focus only on the personality among different clients instead of the personality within a client which widely exists in the real social networks, where intra-client personality addresses the heterogeneity of known data, while inter-client personality always tackle client heterogeneity under privacy constraint. In this paper, we propose a novel automatic personalized graph federated learning (PGFL) scheme named FedCCH to capture both inter-client and intra-client heterogeneity. For intra-client heterogeneity, we innovatively propose the learnable Personalized Factor (PF) to automatically normalize each graph representation within clients by learnable parameters, which weakens the impact of non-IID data distribution. For inter-client heterogeneity, we propose a novel hash-based similarity clustering method to generate the hash signature for each client, and then group similar clients for joint training among different clients. Ultimately, we collaboratively train intra-client and inter-client modules to improve the effectiveness of capturing the heterogeneity of the graph data of clients. Experiment results demonstrate that FedCCH outperforms other state-of-the-art baseline methods.

1 Introduction

Federated Learning (FL) enables distributed machine learning while addressing privacy and security concerns by keeping data locally. [McMahan *et al.*, 2017]. However, in FL, clients often exhibit diverse data distributions, thereby contravening the standard assumption of independent and identically distributed (i.i.d) data prevalent in centralized machine learning. This phenomenon, referred to as data heterogeneity, is recognized as a significant factor contributing to the performance degradation of the global model. Therefore, those non-i.i.d phenomenon drive the development of personalized federated learning (PFL) [Tan *et al.*, 2022], which learns customer-specific personalized models with a global model as an integral part of knowledge sharing [Tan *et al.*, 2022; Gauthier *et al.*, 2023]. PFL satisfies the personalized requirements by delivering tailored model parameters to unique model architectures of diverse devices(same as clients in FL), which is called model customization. Model customization becomes imperative for individual devices or users, particularly for cooperative training in cross-client scenarios such as speech recognition [Sim *et al.*, 2019], anomaly detection [Haotian *et al.*, 2022], and image segmentation [Jiang *et al.*, 2022] among different clients. Recently, GFL has emerged as a dominant approach to address the limitation of Graph Neural Networks (GNNs) [Kipf and Welling, 2017; Velickovic *et al.*, 2017; Xu *et al.*, 2018] when facing scattered graph data in the real world with privacy concerns [Fu *et al.*, 2022]. Existing GFL methods are categorized into three types, differentiated by the graph data each client holds. (1) Node-level separation [Chen *et al.*, 2022; Pan *et al.*, 2023] (2) Subgraph-level separation [Zhang *et al.*, 2021; Chen *et al.*, 2021] and (3) Graph-level separation [Tan *et al.*, 2023; Xie *et al.*, 2021]. Similar to traditional FL, GFL also faces the issue of data heterogeneity, and several studies have been conducted to address related challenges [Xie *et al.*, 2021; Gauthier *et al.*, 2023]. However, they still lack specific considerations for graph data. Particularly in graph-level feder-

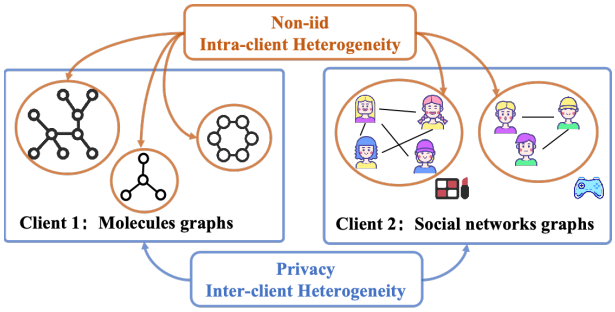

Figure 1: Heterogeneity of the graph data between and within clients. Client 1 exhibits significant structural heterogeneity, while Client 2 demonstrates notable feature heterogeneity.

*Corresponding author: Huaming Wu

ated learning, there is also a certain degree of heterogeneity among the graphs within each client. Those pose significant challenges to the overall goals of federated learning in various graph-based applications. As shown in Fig. 1, molecular structures can exhibit both ring and tree-like configurations, with substantial differences in their underlying attribute distributions.

Additionally, significant domain differences across clients contribute to common forms of heterogeneity. To tackle the challenges caused by the heterogeneity, clustering techniques in personalized federated learning have emerged as a crucial solution. Previous works [Xie *et al.*, 2021; Sattler *et al.*, 2020] have attempted to address these challenges by clustering based on partial graph data or statistical information. However, the former approach compromises user privacy, giving rise to privacy vulnerabilities since sensitive information may be exposed during the clustering stage, while the latter inaccurately represents the data, leading to suboptimal outcomes in federated graph learning. Therefore, enhancing the performance of GFL with heterogeneous issues while ensuring privacy protection presents a significant challenge.

To address the contradiction between privacy and personality widely existed in heterogeneous environments, we propose a novel framework named personalized graph **Fed**erated learning with inter-**C**lient and intra-**C**lient **H**eterogeneity (**FedCCH** for brevity) for personalized graph federated learning with data privacy in heterogeneous environments. The main contributions can be summarized as follows:

- *Joint learning for Intra- and inter-client personalization:* We considered the issues of intra-client and inter-client heterogeneity present in GFL in heterogeneity environments and proposed a novel personalized Graph Federated Learning approach, which is the first work to jointly learn intra- and inter-client personalization.
- *Automatic Personalization Factor for Intra-Heterogeneity:* To address intra-client data heterogeneity, we propose a novel learnable PF by individually normalizing each graph within a client to achieve an approximate data feature distribution and excluding PF from federated learning. It is worth mentioning that FedCCH does not introduce additional hyper-parameters and can automatically learn personality.
- *Hash-based Clustering for Personalization Learning:* To address inter-client heterogeneity of graph data which deteriorates GNNs generalization, we innovatively utilize the hash algorithm to generate hash signatures for similarity calculations and clustering to protect client privacy while capturing heterogeneous relationships between clients.
- *Effective Performance:* We conduct extensive experiments on 15 graph datasets, which include 6 different data combinations to assess the performance of the proposed FedCCH framework. The results surpass multiple FL and GFL algorithms, outperforming other methods by up to 5.04% in average test accuracy on BIO-SN-CV.

2 Preliminaries

2.1 Federated Learning

In FL, each client owns a local private dataset D_i drawn from distribution $P_i(x, y)$, $i = 1, 2, \ldots, m$, where x and y denote the input features and corresponding class labels respectively, and m denotes the number of clients. Usually, clients share a model $\mathcal{F}(\omega; x)$ with the same architecture and hyperparameters. This model is parameterized by learnable weights ω and input features x. The objective function of FedAvg [McMahan *et al.*, 2017] is:

$$\min_{\omega} \sum_{i=1}^{m} \frac{|D_i|}{N} \mathcal{L}_S(\mathcal{F}(\omega; x), y), \tag{1}$$

where ω is the global model's parameters, N is the total number of instances over all clients, $\mathcal{F}$ is the shared model, and $\mathcal{L}_S$ is a general definition of any supervised learning task.

Nonetheless, In the statistical heterogeneity setting in the real world, P_i varies across clients, indicating heterogeneous input/output space for x and y. For example, P_i on different clients can be the data distributions over different subsets of classes. For the i-th client, the training procedure is to minimize the loss as defined below:

$$\min_{\omega_1, \omega_2, \ldots, \omega_m} \sum_{i=1}^{m} \frac{|D_i|}{N} \mathcal{L}_S(\mathcal{F}_i(\omega_i; x), y). \tag{2}$$

Most existing methods cannot well handle the heterogeneous settings above. Thus, the global model's parameter ω cannot be optimized by averaging $\omega_1, \omega_2, \ldots, \omega_m$. In a GFL framework, ω_i and D_i refer to the parameter set of GNN model and the graph dataset at client i, respectively.

2.2 Graph Neural Networks

Let $G = (V, E)$ be a graph consisting of a set of nodes V and a set of edges E connecting nodes. Each node $v \in V$ has a feature vector $\mathbf{x}_v$. Based on the graph structure and node features, GNNs can be used to learn the node-level representation vector $\mathbf{h}_v$ of node $v \in V$ and/or the graph-level representation vector $\mathbf{h}_G$ of graph G. Existing GNNs usually follow the message passing scheme where $\mathbf{h}_v$ is iteratively updated by aggregating the representations of node v's neighbors. l-th layer in L-layer GNNs can be formulated as:

$$\mathbf{a}_v^{(l)} = \mathrm{AGGREGATE}^{(l)} \left(\left\{ \mathbf{h}_u^{(l-1)} : u \in \mathcal{N}(v) \right\} \right), \tag{3}$$

$$\mathbf{h}_v^{(l)} = \mathrm{UPDATE}^{(l)} \left(\mathbf{h}_v^{(l-1)}, \mathbf{a}_v^{(l)} \right), \tag{4}$$

where $\mathbf{h}_v^{(l)}$ is the representation vector of node v output by the l-th layer, $\mathcal{N}(v)$ is the set of node v's neighbors. Different AGGREGATE and UPDATE strategies are adopted in different GNN variants and sometimes can be integrated together [Kipf and Welling, 2017].

Specifically, for graph classification, the graph representation $\mathbf{h}_G$ can be further obtained by aggregating all the node representations involved in graph G via various graph-level readout functions, such as summation and mean pooling.

3 Related Work

3.1 Federated Learning

FL allows isolated data to collaboratively train a global model with strong generalization ability. FedAvg [McMahan *et al.*, 2017], a classic FL algorithm, performs local training on clients iteratively and averages global parameters on a central server for the next round of client training. Various methods have been proposed to optimize communication cost [Hamer *et al.*, 2020; Wu *et al.*, 2022], data privacy [Wei *et al.*, 2023; Tang *et al.*, 2023], and model accuracy [Wu *et al.*, 2023; Niu and Deng, 2022] in FL. This paper primarily focuses on enhancing model accuracy, which is independent and complementary to other methods concerning communication cost and data privacy. However, the heterogeneity of data distribution can negatively impact the generalization performance of the federated learning global model on each client.

To address this issue, PFL is proposed, which can be divided into data-based PFL and model-based PFL. Data-based PFL aims to reduce heterogeneity among clients by sharing part of the global data [Zhao *et al.*, 2018], but it may pose privacy risks. To mitigate this, model-based PFL was introduced. Model-based PFL can be further categorized into single-model [Li *et al.*, 2020; Karimireddy *et al.*, 2020; Wang *et al.*, 2019; Yue *et al.*, 2021] and multi-model PFL approaches [Dinh *et al.*, 2020; Arivazhagan *et al.*, 2019]. However, the inherent characteristics of graph data introduce additional layers of heterogeneity that exacerbate the challenges in federated learning. This includes the variability in degree distribution across different graphs, the varying levels of sparsity within these graphs, and the diversity in graph feature representations. Such complexities render traditional federated learning methods less effective.

3.2 Graph Federated Learning

The rapid development of FL, along with the benefits of processing graph data using GNNs, has fostered the emergence of GFL, which enables the processing of distributed graph data through FL. Previous work in this area can be classified into three categories:

In node-level graph federated learning, each client possesses an ego graph, which comprises links to other clients, often encapsulating a wealth of information. This approach utilizes the client-composed graphs as the primary objects of learning, aiming to enhance the overall model's capabilities. SFL in [Baek *et al.*, 2023] learns both the global and personalized models simultaneously using client-wise relation graphs and clients' private data to enhance the knowledge-sharing process in PFL. Lumos in [Pan *et al.*, 2023] design a novel tree constructor to improve the representation capability given the limited structural information.

In subgraph-level graph federated learning, where the client only holds a part of the complete graph data [Zheng *et al.*, 2021; Chen *et al.*, 2021; Zhang *et al.*, 2021; Zhang *et al.*, 2022; Liu *et al.*, 2021; Baek *et al.*, 2023; Guo *et al.*, 2023]. These works address information loss after splitting subgraphs, either by generating missing nodes [Zhang *et al.*, 2021] or improving federation strategies [Zheng *et al.*, 2021]. This research is mainly oriented toward node classification or link prediction tasks on graphs.

In graph-level graph federated learning, each client holds multiple complete graphs and mainly uses graph classification as the main downstream task [Hu *et al.*, 2022; Xie *et al.*, 2021; Tan *et al.*, 2023]. Existing research mainly focuses on the heterogeneity of graphs between different clients, GCFL in [Xie *et al.*, 2021] performance clustering federated learning based on gradients. FedStar in [Tan *et al.*, 2023] tries to share domain-agnostic structural information to enhance local model performance. However, existing methodologies often overlook the heterogeneity prevalent within client-specific graph data, notably in aspects such as graph degree distribution and feature distribution. This oversight consequently restricts the performance efficacy of Graph Neural Networks (GNNs). Our research is directed towards developing an advanced graph federation architecture. This architecture is meticulously designed to proficiently address both intra-client and inter-client heterogeneity without additional privacy concerns compared to traditional methods.

4 Methodology

4.1 FedCCH Overview

In our federated learning framework, we consider a scenario with m distinct clients, each identified by indices $1, 2, \cdots, m$. These clients operate under the oversight of a semi-honest or curious server. Each client possesses a collection of complete graphs, with all graphs from a single client belonging to the same domain. Since graph data in different domains may contain complementary information. Through federated learning, various fields can share their domain expertise, thereby improving the performance of the overall model. For instance, client A, representing a biological company, holds extensive biological graph data, while client B, representing a chemical company, maintains a substantial repository of chemical molecular graphs. The objective is to collaboratively train multiple Graph Neural Network (GNN) models that fulfill the requirements of all participating entities. Finally, the system trains each client's model based on local graph data and performs clustering federated average learning. We can formulate the goal of the system as Eq 2.

4.2 Automatic Personalization Factor

We introduce 'intra-client heterogeneity' as a novel concept, specifically referring to the variability in structural and feature compositions among different graphs within the same client. To address this challenge, we advocate using feature standardization and transformation techniques, such as BatchNorm and LayerNorm.

However, this one-size-fits-all approach may fail to address the unique characteristics of each client's data effectively. Secondly, the necessity of model sharing in a federated environment raises concerns, particularly with regularization layers that encapsulate extensive information, potentially leading to privacy vulnerabilities.

To mitigate these issues, we propose the introduction of personalization factors. This concept is designed to navigate

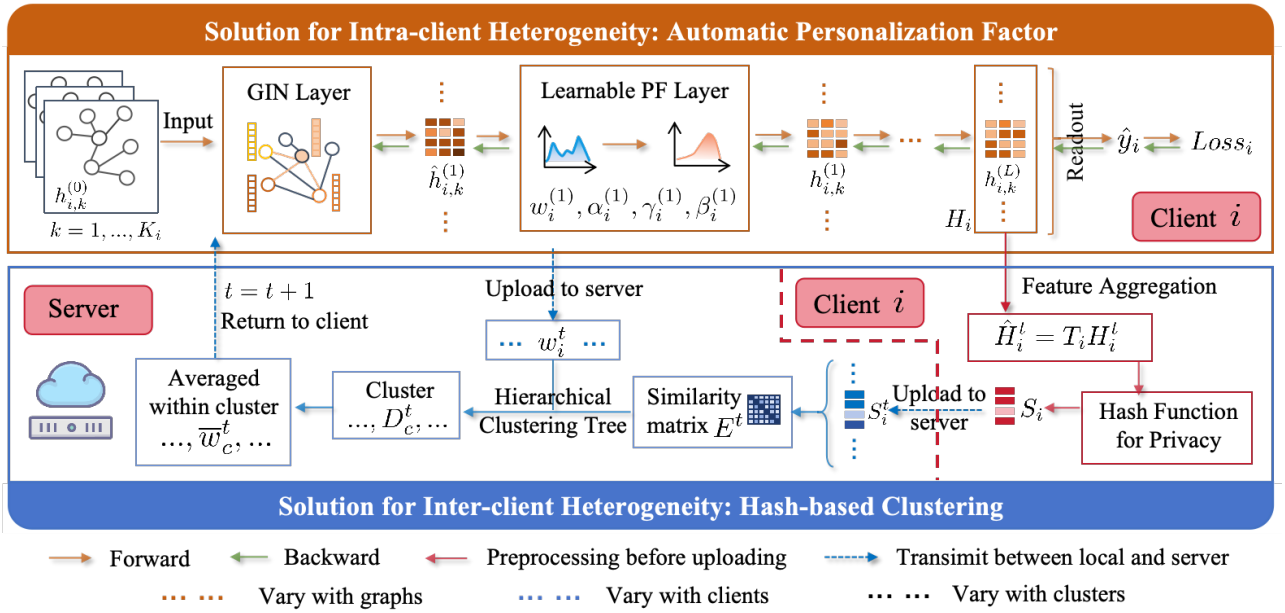

Figure 2: An overview of the GFL architecture in our proposed FedCCH. The top section represents each client and conducts local GNNs training, obtaining a unique signature through readout and hash. The client then uploads this signature, along with its model parameters ω_i^t to the server. The left-bottom section corresponds to the server. The server calculates the similarity of the client's signature, performs clustering, and utilizes the federated average algorithm to share knowledge within each cluster.

the balance between standardization and customization. By applying these factors, local data undergoes feature normalization tailored to the specific characteristics of each client's dataset. Crucially, while standardizing local data, the personalization factors refrain from sharing any federated data. This approach aims to preserve data privacy while ensuring that the normalization process is more closely aligned with the intrinsic properties of each client's data, thereby enhancing model relevance and effectiveness in a federated learning context.

Inspired by [Cai *et al.*, 2021], considering the particularity of the graph structure, structural information for classification will be lost when removing the norm mean. we have elected to employ it as our preferred standardization method in this study. We perform the GIN layer to obtain the hidden embedding for each graph G in the client i as follows:

$$\hat{h}_{i,k}^{(l)} = \mathrm{GIN}(h_{i,k}^{(l-1)}), \tag{5}$$

where $l = 1, \ldots, L$, $k = 1, \ldots, K_i$, and K_i is the number of graphs in the i-th client. $h_{i,k}^{(0)}$ is the input graph, $\hat{h}_{i,k}^{(l)}$ denotes the hidden embedding of the graph in the l-th layer. $GIN(\cdot)$ denotes the GIN layer [Xu *et al.*, 2018] of the model. We denote the element of v-th row and d-th column in $h_{i,k}^{(l)}$ as $h_{i,k,v,d}^{(l)}$, which is abbreviated as $h_{v,d}^{(l)}$ for the purpose of simplification when computing PF factors. The value of PF factor is computed as follows:

$$\mathrm{PF}\left(h_{v,d}^{(l)}\right) = \gamma_{i,d}^{(l)} \cdot \frac{h_{v,d}^{(l)} - \alpha_{i,d}^{(l)} \cdot \mu_d}{\hat{\sigma}_d} + \beta_{i,d}^{(l)}, \tag{6}$$

where $d = 1, 2, \ldots, d^{(l)}$ represents the index of the embedding of the l-th layer, $\mu_d = \frac{\sum_{v=1}^{n} h_{v,d}^{(l)}}{n}$, $\hat{\sigma}_d^2 = \frac{\sum_{v=1}^{n} \left(h_{v,d}^{(l)} - \alpha_d^{(l)} \cdot \mu_d\right)^2}{n}$, $\gamma_{i,d}^{(l)}, \beta_{i,d}^{(l)}$ are the affine parameters as BatchNorm, $h_{v,d}$ specializes in the d-th parameter of the v-th node in the graph, and $\alpha_{i,d}^{(l)}$ is the learnable parameter that plays a crucial role in controlling the retention of information and enhancing the expressive capability of GNNs.

The retention of PF at the local client level, while excluding them from the federated learning's shared parameters, is a strategic approach that caters to the specific needs of individual graph datasets. This methodology permits each set of graph data within a client to adapt to its unique distributional characteristics, significantly enhancing the suitability and efficacy of the model for that particular data. After the PF and ReLU [Glorot *et al.*, 2011], we get the embedding of the l-th layer as follows:

$$h_{i,k}^{(l)} = \mathrm{ReLU}(\mathrm{PF}(\hat{h}_{i,k}^{(l)})), \tag{7}$$

After the transformation of L layers of the neural network, the final representation $h_{i,k}^{(L)}$ is obtained. Next, the embedding is processed by the READOUT function which is summation to obtain the entire graph's representation h_k and compared with the ground truth $\hat{y}_i$ label to calculate the loss and perform gradient descent. The loss function is Cross-Entropy loss as follows:

$$Loss(y_i, \hat{y}_i) = -\sum_{r=1}^{R} y_{i,r} \log(\hat{y}_{i,r}), \tag{8}$$

where R represents the number of graph categories. It is worth noting that parameters such as $\alpha_i^{(l)}, \gamma_i^{(l)} and \beta_i^{(l)}$ are updated after the gradient is backpropagated. The specific process is shown in the upper part of Fig 2.

As a result, local models are capable of achieving faster convergence rates, effectively addressing the challenge of intra-client graph data heterogeneity. The implementation of the PF plays a pivotal role in ensuring that disparate graph datasets within a single client can approximate and learn from their respective distributional representations. Consequently, this leads to an overall improvement in model performance, substantially mitigating the effects of data heterogeneity present within individual clients.

4.3 Hash-based Clustering

In real-world settings, the inherent data heterogeneity among different clients presents a significant challenge to the effectiveness of traditional federated learning methods. To address this, clustering techniques in personalized federated learning have emerged as a key solution. These techniques involve grouping clients based on the similarity in their data distributions, allowing for model training to be tailored to each cluster. This strategy enhances the relevance and suitability of the model to the particular characteristics of the clients within a given group. Implementing clustering in federated learning enhances model performance by tailoring each model to the specific characteristics of its cluster's data, leading to increased accuracy and efficiency. Additionally, it optimizes resource utilization by focusing training and communication on similar clients, conserving computing and communication resources across the network.

While client-side clustering in federated learning presents several advantages, it is not devoid of challenges. Firstly, though this approach can enhance privacy protection, the clustering process often necessitates a degree of data sharing or analysis. This requirement could potentially lead to privacy vulnerabilities, as sensitive information might be exposed during the clustering phase. Secondly, the reliance on indirect features in certain clustering algorithms might not accurately represent the extent of heterogeneity across diverse data distributions. Such algorithms might fail to capture the nuanced differences between datasets, thereby limiting the effectiveness of the clustering. Lastly, the dynamic nature of data in real-world scenarios poses another challenge. Static clustering methods may struggle to adapt to changes in similarity measures over time, leading to outdated or suboptimal groupings.

We adopt aggregation and hash operations to preserve essential client features while ensuring data privacy to tackle this challenge and explore the potential of low-dimensional data representations. Considering the data scale drift among different datasets, we aim to streamline computations and minimize communication costs by achieving a uniform length representation for all clients. To achieve this, we get the final embedding $h_{i,k}^L$ of graph k at last round of epochs in the client, and aggregate the all graph's embedding as $H_i^t = [h_{i,1}^L, \ldots, h_{i,K_i}^L]$, where t represents the number of FL training rounds. Then we compute the unified representation of client i at t-round after aggregation as follows:

$$\hat{\mathbf{H}}_i^t = \mathbf{T}_i \mathbf{H}_i^t, \tag{9}$$

where $\mathbf{T}_i = \frac{1}{|D_i|} \mathbf{1}^T$ represents the aggregation matrix for mean-pooling and $|D_i|$ is the size of graphs of i-th client.

Indeed, hash-based similarity computation has gained popularity in the field of computer vision for achieving faster computation. This approach involves converting two images into a consistent range of pixels and utilizing a hash function to generate a signature that represents the images' similarity. By using this compact representation, image similarity can be efficiently calculated, making it an effective method for large-scale image processing tasks. Hash-based image similarity computation has been widely adopted in various computer vision applications, including image retrieval, object detection, and image clustering, due to its computational efficiency and satisfactory performance.

While most algorithms designed for Euclidean data might not adapt to the graph data, we leverage the low-dimensional representation of the data as it reflects valuable information. For our approach, we consider the low-dimensional vector matrix generated by the client as the input, treating it as Euclidean data for the algorithm. In the last round of GNNs training, the resulting embedding is compressed into a matrix with consistent dimensions, which serves as the input for the hash function as follows:

$$\mathbf{S}_i^t = HASH(\hat{\mathbf{H}}_i^t), \tag{10}$$

where $HASH$ represents a special mapping function that maps $\hat{\mathbf{H}}_i^t$ to $\mathbf{S}_i^t$. After the clients send their hash signatures to the server, the server calculates the similarity $\mathcal{E}_{ij}^t$ between client i and client j based on the corresponding hash signatures. The specific formula is as follows:

$$\mathcal{E}_{ij}^t = 1 - \text{HammingDistance}(\mathbf{S}_i^t, \mathbf{S}_j^t)/d^l, \tag{11}$$

where HammingDistance$(\cdot)$ is employed to calculate the dissimilarity between two hash signatures. It involves comparing each bit of the vector to determine if they are the same. If they differ, the Hamming Distance is incremented by one. By summing up the differences in the bits, we obtain the HammingDistance as a measure of dissimilarity.

The similarity between client i and client j is obtained by subtracting the Hamming distance by one and dividing it by the length of the hash signature. Based on the similarity values, a similarity matrix E is formed as follows:

$$E_{ij}^t = \begin{cases} \mathcal{E}_{ij}^t, & \text{if } i = j \\ 1, & \text{otherwise} \end{cases} \tag{12}$$

Based on the computed similarity matrix, we employ commonly used clustering methods, such as k-means and Hierarchical Clustering Tree (HCT), for the clustering process. In this paper, we primarily utilize HCT as the clustering method to group the clients into multiple clusters based on their similarity. After determining the clustering result, the client assigned to a specific cluster will engage in synchronous training within that cluster. This involves performing FL specifically within the assigned cluster.

4.4 Model Training Process

FedCCH combines PF with clustering federated learning to enable individual clients to learn knowledge more effectively and achieve better training results in the presence of heterogeneous graph data. Fig. 2 illustrates the overall training

# datasets	CHEM		BIO-CHEM		MIX		BIO-SN-CV		BIO-CV		ALL	
# client num	7		9		12		8		5		15	
Accuracy	avg.	gain	avg.	gain	avg.	gain	avg.	gain	avg.	gain	avg.	gain
Local	75.19±1.05	-	71.67±2.37	-	69.70±2.02	-	66.34±3.72	-	70.48±1.48	-	71.69±2.64	-
FedAvg	76.23±2.81	1.04	70.72±2.73	-0.96	69.34±2.75	-0.35	66.25±3.21	-0.09	71.25±2.00	0.07	69.60±2.43	-2.09
FedProx	74.88±2.14	-0.31	71.28±2.40	-0.40	69.92±2.63	0.22	67.48±2.41	1.14	70.47±2.17	-0.01	69.55±2.25	-2.14
FedBN	76.48±4.94	1.29	69.88±4.86	-1.79	69.07±3.05	-0.62	65.44±2.69	-0.90	69.79±2.25	-0.68	67.47±5.03	-4.22
GCFL	75.63±2.18	0.44	72.41±3.53	0.73	70.56±1.97	0.86	67.77±2.50	1.43	69.58±1.93	-0.90	72.60±1.83	0.91
FedStar	77.30±2.59	2.11	74.31±2.08	2.64	70.32±3.38	0.62	66.99±3.02	0.65	66.94±1.88	-3.54	72.02±2.74	0.33
FedCCH-PF	77.42±2.04	1.73	73.40±2.24	2.23	70.87±1.98	1.18	70.90±2.14	4.56	74.03±1.69	3.55	72.39±1.51	0.70
FedProx+PF	77.34±3.07	2.15	71.57±2.48	-0.10	69.68±2.41	-0.02	68.02±2.37	1.68	72.25±1.96	1.77	71.87±2.37	0.18
FedAvg+PF	77.71±2.76	2.52	72.96±3.81	1.28	70.06±2.13	0.36	67.75±1.87	1.41	71.07±1.84	0.59	70.90±2.27	-0.79
FedCCH	**78.17±2.33**	**2.98**	**74.77±2.71**	**3.09**	**72.97±0.98**	**3.28**	**71.38±1.81**	**5.04**	**74.57±2.01**	**4.01**	**75.00±2.21**	**3.31**

Table 1: Performance on different federated graph classification tasks under multiple datasets owned by different clients. Underlining represents the optimal approach without considering intra-client graph heterogeneity.

pipeline of our proposed method, where each client, regardless of whether they have the same or different domain graph data, employs the same model architecture.

From left to right, for the client i, the model is trained using its local data. Subsequently, the client computes the signature of its data, uploads the signature information along with the model, and retains the PF parameters locally. This process ensures that each client can effectively personalize their training process while still contributing to the overall FL framework. After clustering in the server and dividing into c classes, the model parameters are averaged within the classes, and the averaging step is similar to FedAvg [McMahan *et al.*, 2017], which behaves as follows:

$$\overline{w}_c^t = \sum_{i \in C} \frac{|D_i|}{|D_c|} w_i^t. \tag{13}$$

where $|D_i|$ refers to the number of graphs in the local dataset of client i and $|D_c|$ refers to the total number of graphs across all clients in the c-th cluster. Then, the server returns $\overline{w}_c^t$ to clients in the c-th cluster, and clients update their local model parameters w_i^{t+1} by $\overline{w}_c^t$ except for the PF and start local training for the next round. The complete algorithm is shown in the Appendix.

5 Experiments

5.1 Experimental Settings

Datasets We use 15 public graph classification datasets from four different fields, including small molecules (MUTAG, BZR, COX2, DHFR, PTC_MR, AIDS, NCI1), bioinformatics (ENZYMES, DD), social network (COLLAB, IMDB-BINARY, IMDB-MULTI) and computer vision (Letter-low, Letter-high, Letter-med)[Morris *et al.*, 2020].To thoroughly assess the method's effectiveness on heterogeneous graph data, we segmented the aforementioned graph data into six different combinations as CHEM, BIO-CHEM, MIX, BIO-SN-CV, BIO-CV, ALL. Specific information on all graph data can be obtained from the Appendix. In each setting, the client owns the corresponding data set One, by default is randomly divided into three parts: 80% for training, 10% for validation, and 10% for testing.

5.2 Baseline and Experiment Setting

We compare FedCCH with eight baselines. It includes classic federated learning algorithms **FedAvg**[McMahan *et al.*, 2017], **FedProx** [Li *et al.*, 2020] and **FedBN** [Li *et al.*, 2021]. graph federated learning algorithms **FedStar** [Tan *et al.*, 2023] and **GCFL** [Xie *et al.*, 2021], and specially modified classic federated learning algorithms **FedAvg+PF** and **FedAvg+PF**. The details of these baselines are provided in the Appendix. For model, we primarily utilize a three-layer Graph Isomorphism Network (GIN) as our model. Before the activation function, we incorporate Batch Normalization and Graph Normalization layers. The hidden layer dimension of the GIN is set to 64. For the local training process, we employ 1 epoch and a batch size of 64. The optimization is performed using the Adam optimizer with a weight decay of 5×10^{-5} and a learning rate of 0.001. Regarding the FL settings, we set the number of communication rounds to 200 for all FL methods. We employ the Ahash method for hashing and a hierarchical clustering tree for clustering. To ensure robustness in the experimental results, we conduct 5 experiments with different random seeds and calculate the mean and variance. The experiments are conducted on a system equipped with an Nvidia 3090Ti GPU configuration.

5.3 Experimental Results

Performance on Graph Classification. We show federated graph classification results for all methods under six non-IID settings, including a cross-dataset setting (CHEM) and five cross-domain different settings (BIO-CHEM, BIO-CHEM-SN, BIO-SN-CV). We summarize the final average test accuracy and its average gain compared to Local in Ta-

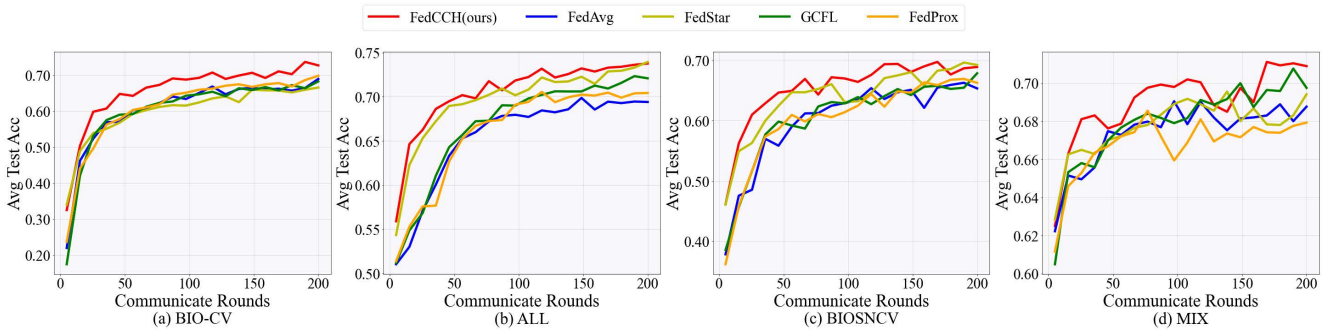

Figure 3: Test accuracy curves of our FedCCH and five FL/FGL methods along the communication rounds.

ble 1. The results show that FedCCH is significantly better than all baselines overall, and the effect is even more obvious after combining the proposed PF.

Among them, traditional FL methods (such as FedAvg and FedProx) cannot surpass Local due to the inevitable performance deterioration in non-IID settings. In particular, FedAvg dropped by 0.96% and 2.09% in BIO-CHEM and ALL respectively, while FedProx dropped by 0.40% and 2.14% in BIO-CHEM and ALL respectively. FedBN, which is designed for personalization, is not suitable for graph data due to its BatchNorm layer and performs poorly in all datasets, losing 4.22% accuracy in the ALL dataset. The two FGL methods (FedStar and GCFL) perform better than Local in most cases. The structure embedding sharing strategy adopted by FedStar aims to share feature-independent structural information, effectively avoiding performance degradation caused by feature heterogeneity and with a 2.64% accuracy increase on BIO-CHEM, but in some datasets such as BIO-CV, the effect is unsatisfactory due to potential heterogeneity in the structure of the data. GCFL uses the nature of gradient information to cluster data information and outperforms Local with a 1.43% accuracy increase on BIO-SN-CV, but due to the limitations of gradient information, most of the effects are not as significant as FedCCH.

Convergence Analysis Fig. 3 illustrates the average test accuracy curves of different FL methods on four datasets, excluding the baseline with regularization technology. It can be observed that across all datasets, FedCCH exhibits faster convergence speed and achieves better average test accuracy. Compared with other baselines, FedCCH saves 100 rounds of training time on average, which is attributed to the fact that the hash clustering method captures more intrinsic similarities between the data, enabling each client to achieve improved performance.

5.4 Ablation Study

Ablation Experiment of Hash Mechanisms. In our experiment, we removed the hash mechanism and solely utilized cosine similarity, calculated by embedding the data, to represent the similarity between clients and clusters. As depicted in Fig. 4, when compared with the method without the hash mechanism, the performance of FedCCH has not significantly declined. In fact, there is a certain improvement observed in most of the datasets, which indicates that the hash clustering mechanism can contribute to the accuracy performance of the model while reducing the sensitivity of the data.

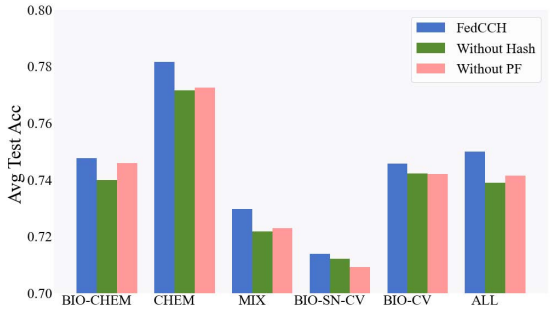

Figure 4: The average test accuracy and variance of FedCCH for ablation experiments on hash mechanisms and PF on different datasets.

Ablation Experiment of The PF To investigate the impact of the PF on the overall performance of FL, we conducted experiments on multiple datasets using FedCCH by sharing its layer globally. As depicted in Fig. 4, there is a substantial improvement in performance when retaining the PF locally across different datasets, which suggests that the PF is better suited for local clients to undergo specific training without the need for sharing, resulting in reduced communication costs in FL.

6 Conclusion

In this paper, we define the intra-client heterogeneity and inter-client heterogeneity that exist in graph-level graph federation learning and propose FedCCH to overcome the training challenge caused by these two kinds of heterogeneity. For details, the proposed framework FedCCH improve the training performance by collaboratively training two novel modules: one novel module is hash-based similarity clustering and the other novel module is the learnable PF normalizing, where hash-based similarity clustering can capture inter-client heterogeneity and protect client privacy, while PF normalizing can capture intra-client heterogeneity and bring a faster convergence. Through extensive experiments on 15 graph datasets, the results demonstrate the effectiveness and superiority of the proposed FedCCH. In future work, we will study how to improve the robustness of graph federated systems while handling graph data heterogeneity.

Acknowledgments

This work was supported in part by the Zhejiang Provincial Natural Science Foundation of China under Grant LDT23F01012F01 and LQN25F020016 and by the National Natural Science Foundation of China under Grant Number 62401190, 62372146 and 62071327.

References

[Arivazhagan *et al.*, 2019] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *CoRR*, abs/1912.00818, 2019.

[Baek *et al.*, 2023] Jinheon Baek, Wonyong Jeong, Jiongdao Jin, Jaehong Yoon, and Sung Ju Hwang. Personalized subgraph federated learning. In *International Conference on Machine Learning*, pages 1396–1415. PMLR, 2023.

[Cai *et al.*, 2021] Tianle Cai, Shengjie Luo, Keyulu Xu, Di He, Tie-yan Liu, and Liwei Wang. Graphnorm: A principled approach to accelerating graph neural network training. In *International Conference on Machine Learning*, pages 1204–1215. PMLR, 2021.

[Chen *et al.*, 2021] Fahao Chen, Peng Li, Toshiaki Miyazaki, and Celimuge Wu. Fedgraph: Federated graph learning with intelligent sampling. *IEEE Transactions on Parallel and Distributed Systems*, 33(8):1775–1786, 2021.

[Chen *et al.*, 2022] Fengwen Chen, Guodong Long, Zonghan Wu, Tianyi Zhou, and Jing Jiang. Personalized federated learning with a graph. In *IJCAI*, pages 2575–2582. ijcai.org, 2022.

[Dinh *et al.*, 2020] Canh T. Dinh, Nguyen Hoang Tran, and Tuan Dung Nguyen. Personalized federated learning with moreau envelopes. In *NeurIPS*, 2020.

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

[Gauthier *et al.*, 2023] François Gauthier, Vinay Chakravarthi Gogineni, Stefan Werner, Yih-Fang Huang, and Anthony Kuh. Personalized graph federated learning with differential privacy. *CoRR*, abs/2306.06399, 2023.

[Glorot *et al.*, 2011] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In Geoffrey Gordon, David Dunson, and Miroslav Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 315–323, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR.

[Guo *et al.*, 2023] Jiayan Guo, Shangyang Li, and Yan Zhang. An information theoretic perspective for heterogeneous subgraph federated learning. In *International Conference on Database Systems for Advanced Applications*, pages 745–760. Springer, 2023.

[Hamer *et al.*, 2020] Jenny Hamer, Mehryar Mohri, and Ananda Theertha Suresh. Fedboost: A communication-efficient algorithm for federated learning. In *International Conference on Machine Learning*, pages 3973–3983. PMLR, 2020.

[Haotian *et al.*, 2022] WANG Haotian, ZHENG Dongyi, LIU Fang, and XIAO Nong. Personalized federated anomaly detection method for multivariate time series data. *Journal of Computer Engineering & Applications*, 58(11), 2022.

[Hu *et al.*, 2022] Kai Hu, Jiasheng Wu, Yaogen Li, Meixia Lu, Liguo Weng, and Min Xia. Fedgcn: Federated learning-based graph convolutional networks for non-euclidean spatial data. *Mathematics*, 10(6):1000, 2022.

[Jiang *et al.*, 2022] Meirui Jiang, Zirui Wang, and Qi Dou. Harmofl: Harmonizing local and global drifts in federated learning on heterogeneous medical images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 1087–1095, 2022.

[Karimireddy *et al.*, 2020] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning*, pages 5132–5143. PMLR, 2020.

[Kipf and Welling, 2017] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[Li *et al.*, 2020] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020.

[Li *et al.*, 2021] Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou. Fedbn: Federated learning on non-iid features via local batch normalization. *arXiv preprint arXiv:2102.07623*, 2021.

[Liu *et al.*, 2021] Tao Liu, Peng Li, and Yu Gu. Glint: Decentralized federated graph learning with traffic throttling and flow scheduling. In *2021 IEEE/ACM 29th International Symposium on Quality of Service (IWQOS)*, pages 1–10. IEEE, 2021.

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

[Morris *et al.*, 2020] Christopher Morris, Nils M Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. Tudataset: A collection of benchmark datasets for learning with graphs. *arXiv preprint arXiv:2007.08663*, 2020.

[Niu and Deng, 2022] Yifan Niu and Weihong Deng. Federated learning for face recognition with gradient correction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 1999–2007, 2022.

[Pan *et al.*, 2023] Qiying Pan, Yifei Zhu, and Lingyang Chu. Lumos: Heterogeneity-aware federated graph learning over decentralized devices. *arXiv preprint arXiv:2303.00492*, 2023.

[Sattler *et al.*, 2020] Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE transactions on neural networks and learning systems*, 32(8):3710–3722, 2020.

[Sim *et al.*, 2019] Khe Chai Sim, Petr Zadrazil, and Françoise Beaufays. An investigation into on-device personalization of end-to-end automatic speech recognition models. *arXiv preprint arXiv:1909.06678*, 2019.

[Tan *et al.*, 2022] Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. Towards personalized federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–17, 2022.

[Tan *et al.*, 2023] 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.

[Tang *et al.*, 2023] Xinyu Tang, Cheng Guo, Kim-Kwang Raymond Choo, and Yining Liu. An efficient and dynamic privacy-preserving federated learning system for edge computing. *IEEE Transactions on Information Forensics and Security*, 2023.

[Velickovic *et al.*, 2017] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, Yoshua Bengio, et al. Graph attention networks. *stat*, 1050(20):10–48550, 2017.

[Wang *et al.*, 2019] Kangkang Wang, Rajiv Mathews, Chloé Kiddon, Hubert Eichner, Françoise Beaufays, and Daniel Ramage. Federated evaluation of on-device personalization. *arXiv preprint arXiv:1910.10252*, 2019.

[Wei *et al.*, 2023] Kang Wei, Jun Li, Chuan Ma, Ming Ding, Wen Chen, Jun Wu, Meixia Tao, and H Vincent Poor. Personalized federated learning with differential privacy and convergence guarantee. *IEEE Transactions on Information Forensics and Security*, 2023.

[Wu *et al.*, 2022] Chuhan Wu, Fangzhao Wu, Lingjuan Lyu, Yongfeng Huang, and Xing Xie. Communication-efficient federated learning via knowledge distillation. *Nature communications*, 13(1):2032, 2022.

[Wu *et al.*, 2023] Xueyang Wu, Hengguan Huang, Youlong Ding, Hao Wang, Ye Wang, and Qian Xu. Fednp: Towards non-iid federated learning via federated neural propagation. 2023.

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

[Xu *et al.*, 2018] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

[Yue *et al.*, 2021] Sheng Yue, Ju Ren, Jiang Xin, Sen Lin, and Junshan Zhang. Inexact-admm based federated meta-learning for fast and continual edge learning. In *Proceedings of the Twenty-second International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, pages 91–100, 2021.

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

[Zhang *et al.*, 2022] Chenhan Zhang, Shiyao Zhang, Shui Yu, and JQ James. Graph-based traffic forecasting via communication-efficient federated learning. In *2022 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 2041–2046. IEEE, 2022.

[Zhao *et al.*, 2018] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

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