Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.076676

ARTICLE

SubPFed: A Personalized Federated Learning Approach with Subgraphs

Jianbin Li[1,*], **Hang Bao**[1] **and Xin Tong**[2]

[1]School of Control and Computer Engineering, North China Electric Power University, Beijing, China
[2]School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing, China

*Corresponding Author: Jianbin Li. Email: lijb87@ncepu.edu.cn

Received: 24 November 2025; Accepted: 09 March 2026; Published: 08 May 2026

ABSTRACT: The proliferation of large-scale graph data has enabled Graph Neural Networks (GNNs) to achieve significant success in domains such as recommender systems, social network analysis, and biomedicine. However, in practical networked environments, particularly in distributed service infrastructures, graph data is often isolated between multiple edge smart devices and cannot be shared due to privacy, making GNN models weak in generalization. Subgraph Federated Learning (SFL) mitigates this challenge by treating local client data as subgraphs of the global graph to decentralized GNN training. Unfortunately, client-side missing edges make GNN model difficult to capture dependency information between subgraphs, and local heterogeneous data hinders global model convergence, thereby limiting the performance of federated GNN model. To address this, we propose SubPFed, a personalized federated learning approach tailored for subgraph-based training. SubPFed computes the functional embeddings of local GNNs using random graph inputs. It then estimates subgraph similarity by weighting these embeddings and the structural information of the overlapping nodes. Finally, a personalized weighted aggregation strategy is designed based on the similarity to enhance representation consistency across clients and mitigate data heterogeneity. Experiments on three real-world graph datasets show that SubPFed consistently outperforms state-of-the-art baselines, improving node classification accuracy by 4.28% to 26.50%. Furthermore, SubPFed demonstrates strong robustness under varying subgraph overlap ratios, underscoring its adaptability and scalability in federated graph learning scenarios.

KEYWORDS: Federated learning; graph neural networks; node classification

1 Introduction

With the convergence of Artificial Intelligence (AI), 5G communications, and edge computing has significantly enhanced the interconnectivity and intelligence of Internet of Things (IoT) systems. This advancement has driven the proliferation of distributed smart devices such as smartphones, smart home appliances, and connected vehicles, resulting in the generation of massive graph data. Graph Neural Networks (GNNs), with their powerful modeling ability for graph data, have demonstrated superior performance in tasks such as node classification, link prediction, and graph embedding. Consequently, they are extensively applied in various practical scenarios, including social network analysis, recommendation systems, and bio-chemistry [1–3]. However, most existing research follows a centralized learning paradigm that assumes access to all graph data, which is unrealistic in practical network systems and service environments. In real-world scenarios, graph data is often distributed across multiple administrative domains or institutions (e.g., inter-hospital diagnostic–treatment networks or cross-platform financial transactions) and cannot be directly shared due to privacy constraints [4]. Additionally, training GNN models on graph data from a single institution results in limited generalization capabilities.

Researchers have integrated federated learning (FL) [5] into GNN training to form graph federated learning (GFL) [6] to mitigating this limitation. While some studies focus on graph-level FL [7,8], this paper emphasizes subgraph federated learning (SFL) for enhanced privacy protection in networked environments. SFL partitions the global graph into subgraphs, where each participant trains a local GNN model using their subgraph data. It learns all local subgraph data by exchanging necessary local model parameters or intermediate results without leaking the original subgraph structure and node features.

Although SFL excels in privacy protection, it still faces challenges of missing cross-client links and cross-domain data heterogeneity. Specifically, each client possesses only a portion of the global graph, and local subgraphs may lack shared edges, preventing them from exchanging connection information with other clients. This limitation leads to suboptimal performance of GNN models trained using vanilla FL algorithm [9]. To address the issue of missing links in subgraphs, existing approaches [4,10,11] extend other subgraphs to local subgraph to supplement connection information. However, these methods involve sharing node information, which compromises data privacy and incurs higher communication costs. Moreover, there exists inevitable data heterogeneity across different client subgraphs. Their graph data collected by different methods are typically non-independent and identically distributed (non-IID) [12], exhibiting substantial differences in structure, features, and task objectives, which may lead to difficult convergence of GNN models. To mitigate it, existing methods [13,14] propose personalized aggregation strategies by leveraging model gradients or output similarities. This established an important paradigm for handling data heterogeneity without sharing raw features. Another method [15] is to build a topology and information extractor on the local client side to achieve better representation capabilities. Regrettably, these approaches focus primarily on differences in node features while neglecting the inherent heterogeneity caused by the structural information of subgraphs.

To address the a forementioned challenges and overcome the limitations of existing methods, we focus on the common elements between subgraphs (i.e., overlapping nodes) and propose a novel approach. Specifically, overlapping nodes refer to those with similar or identical identities in the graph structures across different clients. For example, in the health-care domain, shared patient or disease data may exist among various hospitals. In the social network graph depicted in Fig. 1, public figures on social media serve as overlapping nodes across different user graphs. Their information is typically public [16–18] and can be shared across client-side subgraphs. These overlapping nodes, not only enable cross-client information fusion within a privacy-preserving approach while avoiding user privacy leakage but also act as connection hubs between subgraphs, reflecting the distribution of subgraphs to some extent. Thus, overlapping nodes not only address the missing edge information of the subgraphs, but also their structural information and feature representations can be used to infer potential connections and the degree of heterogeneous distribution among subgraphs.

Based on this insight, we introduce SubPFed, a novel personalized SFL approach. This approach achieves personalized SFL by improving the interrelated local GNNs jointly to tackle data heterogeneity. It captures the structural similarity of subgraphs between different clients through personalized FL based on the weighted distance method of overlapping node degrees, alleviating the problem of missing edge links. The measurement of structural similarity is pre-calculated by the server. Do not share any node features, adjacency lists or model gradients. Furthermore, it integrates the function embeddings of GNNs on random graphs with the weighted distance method based on overlapping node degrees to obtain a comprehensive similarity, thereby evaluating data heterogeneity. Then, it performs weighted averaging of local models for each client to achieve personalized SFL. We evaluated the subgraph FL performance of SubPFed on three real-world graph datasets, considering subgraphs with varying node overlapping ratios. On these subgraphs, SubPFed significantly outperforms the advanced baseline for example Fed-PUB [14] and

FedSG [15]. Further analysis reveals that our weighted distance method based on overlapping node degrees effectively captures the structural similarity of subgraphs between different clients. By comprehensively considering both subgraph structural similarity and GNN function embedding similarity, our approach provides a more holistic quantification of client subgraph data distribution compared to single similarity calculations. In summary, our contributions are as follows:

- We propose SubPFed, a novel SFL approach that leverages overlapping nodes to address missing edge information between subgraphs while achieving personalized learning.
- We design a new weighted aggregation method that combines the functional embedding similarity of local GNNs and the structural information similarity of subgraph data result local weight. It can quantify client data similarity more comprehensively and enhance the robustness in heterogeneous environments.
- We validate the performance of SubPFed on three real datasets with different node overlapping rates, the results show that SubPFed possesses significant superiority over all baseline approaches and less communication costs.

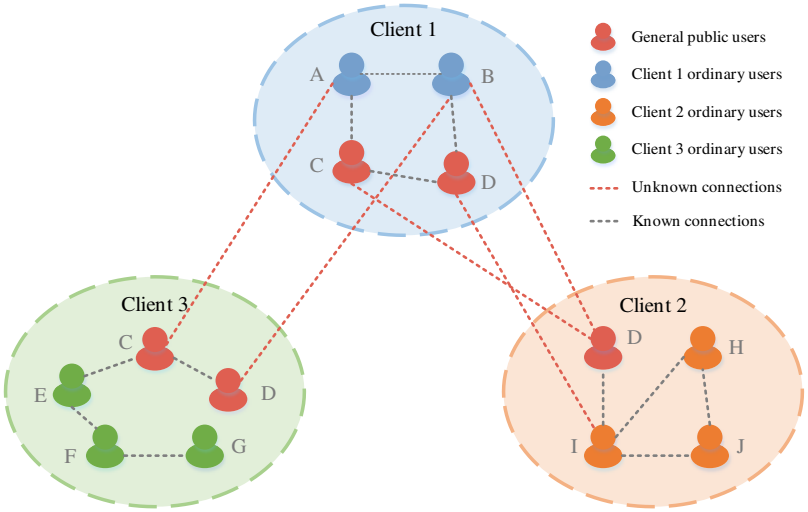


Figure 1: A social network graph with three clients.

2 Related Work

2.1 GNNs

GNNs as a powerful tool, are designed to learn representations of nodes, edges, and the entire graph structure, becoming a critical research direction in graph-structured data analysis [2,6,19]. Most existing GNNs rely on the message-passing neural network framework [20], which iteratively aggregates and updates features from neighboring nodes and itself to enhance the performance of graph models. For instance, Kipf and Welling [21] introduced the Graph Convolutional Network (GCN), performing spectral domain convolutions on node features and approximating the mean aggregation of neighboring nodes. Hamilton et al. [22] proposed GraphSAGE, which generates node embeddings via sampling and aggregation functions. Veličković et al. [23] developed the GAT, incorporating an attention mechanism to dynamically assign importance weights to neighboring nodes. Recently, Fofanah and Leigh [24] proposed the EATSA-GNN model, leveraging edge-aware and two-stage attention mechanisms to improve node classification performance. Despite their success in tasks such as node classification and link prediction, most existing GNNs assume centralized training scenarios and overlook the fact that, in real-world applications, graph

data is often distributed across multiple organizations and cannot be shared due to privacy restrictions, thus limiting their practical applicability.

2.2 FL

FL [5,25] represents an emerging distributed learning paradigm aimed at collaboratively training a global model among multiple participants without sharing raw data. In this framework, participants conduct local training and model updates while relying on a central server for model aggregation, thereby ensuring data privacy. Traditional FL algorithms, such as FedAvg proposed by McMahan et al. [9], aggregate local model parameters to generate a global model. However, as data distribution dis-parities increase, this approach may struggle to effectively address the challenges posed by heterogeneous data. To mitigate this issue, Li et al. [26] introduced FedProx, which incorporates proximal term regularization to reduce discrepancies between local and global models, preventing local models from diverging excessively from local data. Nevertheless, when data heterogeneity is extreme, a single global model may fail to meet the needs of all clients, necessitating personalized client models. Recent studies have also investigated enhancing personalized models through techniques such as knowledge distillation [27], federated meta-learning [28], and adaptive aggregation strategies [29].

2.3 SFL

Existing researches on GFL is primarily categorized into subgraph-level and graph-level [6,30]. Graph-level FL mainly addresses scenarios where different clients possess entirely independent graph datasets, and recent studies [5,31] have predominantly focused on the significant discrepancies in graph labels among clients. In contrast, SFL concentrates on the graph subsets owned by different clients, which are parts of the global graph. Subgraph-level FL encounters unique challenges related to graph structures: each subgraph represents only a portion of the global graph, and there may exist missing edges between subgraphs. In addition, it also faces the challenge of local heterogeneity. To address the issue of missing edges between subgraphs, Wu et al. [10] proposed FedGNN, introducing a cross-client node expansion mechanism that adds relevant nodes from other clients' subgraphs; Zhang et al. [4] proposed FedSage+, which extends local subgraphs by generating missing neighbor information; Yao et al. [11] proposed FedGCN, enhancing nodes by fusing neighbor node features with local node features. These methods all enrich node connection information by requesting node data from other subgraphs, but they incur high communication costs and increase the risk of privacy leakage. To tackle the heterogeneity problem of client subgraphs, Baek et al. [14] introduced Fed-PUB, proposing a personalized SFL approach that adopts personalized aggregation and sparse masking strategies to address the heterogeneity between subgraph communities [32,33]. In addition, Wang et al. [15] proposed FedSG, which builds topologies and information extractors on the local client side to achieve better representation capabilities. On the server side, topological parameters and feature parameters are separated and aggregated to achieve personalized modeling and alleviate data heterogeneity.

However, these approaches overlook the inevitable heterogeneity caused by the structural characteristics of subgraphs themselves, thus failing to fully capture the actual structural differences between subgraphs. SubPFed is a subgraph-level FL approach that comprehensively considers both the structural information of the graph and the functional information of the nodes, leveraging subgraph similarity for personalized federated subgraph learning. This approach avoids direct exchanges of raw data or gradients between clients, thereby addressing the limitations of existing methods. Table 1 provides a clearer comparison between SubPFed and existing methods.

Table 1: The differences between our method and other baseline methods are compared from four dimensions.

	FedAvg	FedGNN	FedGCN	Fed-PUB	FedSG	SubPFed
Considering missing neighbor information	×	√	√	√	√	√
Sharing node information	×	√	×	×	√	×
Functional embeddings for subgraph similarities	×	×	×	√	×	√
Overlapping nodes for subgraph similarities	×	×	×	×	×	√

Facing the potential issues between subgraphs in subgraph-level FL, Table 1 summarizes the comparison between the methods proposed in prior work [14,15] and SubPFed presented in this paper. Overall, SubPFed introduces a weighted distance method based on the degree of overlapping nodes, integrating both the structural information of the graph and the functional information of the nodes while utilizing subgraph similarity for personalized federated subgraph learning. The key advantage of this approach lies in its ability to avoid direct exchanges of complete data or gradients between clients, effectively resolving privacy concerns.

3 Preliminaries

3.1 GNNs

GNNs: An undirected graph is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a set of N nodes and $\mathcal{E}$ is a set of m edges. Additionally, the node feature matrix $X \in \mathbb{R}^{N \times f}$ is introduced, with each row representing an f-dimensional feature vector for a node. An edge from node u to node v is denoted as $(u, v) \in \mathcal{E}$. GNNs [1] represent nodes based on their neighborhoods and themselves as follows:

$$H_v^{l+1} = \mathrm{UPD}^l\left(H_v^l, \mathrm{AGG}^l\left(\left\{H_u^l : \forall u \in \mathcal{N}(v)\right\}\right)\right), \tag{1}$$

where H_v^l represents the feature of node v at l^{th} layer, $\mathcal{N}(v)$ denotes the set of adjacent nodes of node v: $\mathcal{N}(v) = \{u \in \mathcal{V} \mid (u, v) \in \mathcal{E}\}$, AGG aggregates the features of the neighbors of node v, UPD updates the representation of node v by combining its previous representation and the aggregated information from its neighbors. H^l is initialized as X. As the number of network layers increases, the embedding calculation of node v gradually incorporates information from neighbors that are several hops away.

3.2 FL

FL aims to collaboratively train models without sharing local data between clients. Suppose there are M clients, each with its own dataset $\mathcal{D}_m = \{(X_i, Y_i)\}_{i=1}^{N_m}$, where X_i represents data instances and Y_i are the corresponding labels, with N_m is the number of data instances. A common algorithm for FL is FedAvg [9], which follows these steps:

1. Initialization. In the first round, the central server initializes the global model parameters $\overline{\theta}$, and each client m sets its local model parameters $\theta_m^{(0)}$ to $\overline{\theta}^{(0)}$, i.e., $\theta_m^{(0)} \leftarrow \overline{\theta}^{(0)}$ for all m.
2. Local Updates. Each client trains its model on its own local dataset $\mathcal{D}_m$ by minimizing the loss function $\mathcal{L}\left(\mathcal{D}_m; \theta_m^{(0)}\right)$, then updates its local parameters: $\theta_m^{(1)} \leftarrow \theta_m^{(0)} - \eta \nabla \mathcal{L}$, where η is the learning rate.

3. Global Aggregation. After the local training, the server aggregates the local models by weighted averaging based on the size of each client's dataset. The updated global model parameters $\overline{\theta}^{(1)}$ are computed as:

$$\overline{\theta}^{(1)} \leftarrow \frac{1}{N} \sum_{m=1}^{M} N_m \theta_m^{(1)}, \tag{2}$$

where $N = \sum_{m=1}^{M} N_m$ is the total number of data points across all clients. The server then sends the updated global model $\overline{\theta}^{(1)}$ to the selected clients for the next round

This process continues, with local clients performing updates on their data, and the server aggregating the models, until the final round R. The model is updated based solely on parameter exchange, with no direct access to the local data of each client.

4 The SubPFed Approach

4.1 Personalized SFL

We investigate a federated graph learning approach consisting of M clients and a central server $\mathcal{S}$. Each client $\mathcal{C}_i$ possesses a distributed subgraph dataset denoted as $\mathcal{D}_i = \{\mathcal{G}_i, Y_i\} = \{\{\mathcal{V}_i, \mathcal{E}_i, X_i\}, Y_i\}, i = 1, \ldots, M$, where $\mathcal{V}_i$ represent the node set, $\mathcal{E}_i$ is the edge set, X_i denotes the node feature set, and Y_i corresponds to the label set. Among the graphs $\mathcal{D}_i$ and $\mathcal{D}_j$ held by different clients' datasets $\mathcal{G}_i$ and $\mathcal{G}_j$, there exists an overlapping node set $V_{overlap}^{ij}$, meaning that for $\forall i \in \{k\}, \exists j \in \{k\} (i \neq j)$, there exists a shared node between the two graphs, i.e., $V_i \cap V_j = V_{overlap}^{ij} \neq \varnothing$. The task of each local client is node classification, which predicts the node labels of the remaining node set by training GNNs. The objective of existing SFL methods [4,10,11] can be formulated as: $\min_{\overline{\theta}} \sum_{G_i \subseteq G} \mathcal{L}\left(G_i; \overline{\theta}\right)$. However, when seeking a universal parameter set (i.e., $\overline{\theta}$) that applies to all subgraphs, it may not always be feasible to find the optimal parameter set due to the inevitable heterogeneity of graph data across clients. Instead, a suboptimal parameter set is often obtained. To address the inevitable influence of graph data heterogeneity among subgraphs from different clients, we propose a personalized SFL approach. This method conducts personalized optimization for each subgraph rather than relying on a unified global parameter set. Our objectives are defined as follows:

$$\min_{\{\theta_i\}_{i=1}^{M}} \sum_{G_i \subseteq G} \mathcal{L}\left(G_i; \theta_i\right), \theta_i \leftarrow \sum_{j=1}^{M} \alpha_{ij} \theta_j, \tag{3}$$

where α_{ij} represents the weight aggregation coefficient between client i and client j. By assigning larger weights to interrelated subgraphs, collaborative learning among local models can be effectively promoted.

Fig. 2 shows the process of SubPFed. Firstly, when SubPFed uses Metis [34] for subgraph partitioning, it calculates the similarity of graph data distribution based on the degree of overlapping nodes among each client. Then, by using random graphs as input for each client, it calculates the similarity between the function embeddings of the local GNNs. And use the comprehensive similarity to perform weighted aggregation of local model parameters when aggregating on the server side. Overall, SubPFed achieves personalized FL of subgraphs in two aspects to solve the heterogeneous problem of graph data: functional embedding similarity and graph structure distribution similarity. The symbols and related definitions are shown in Table 2.

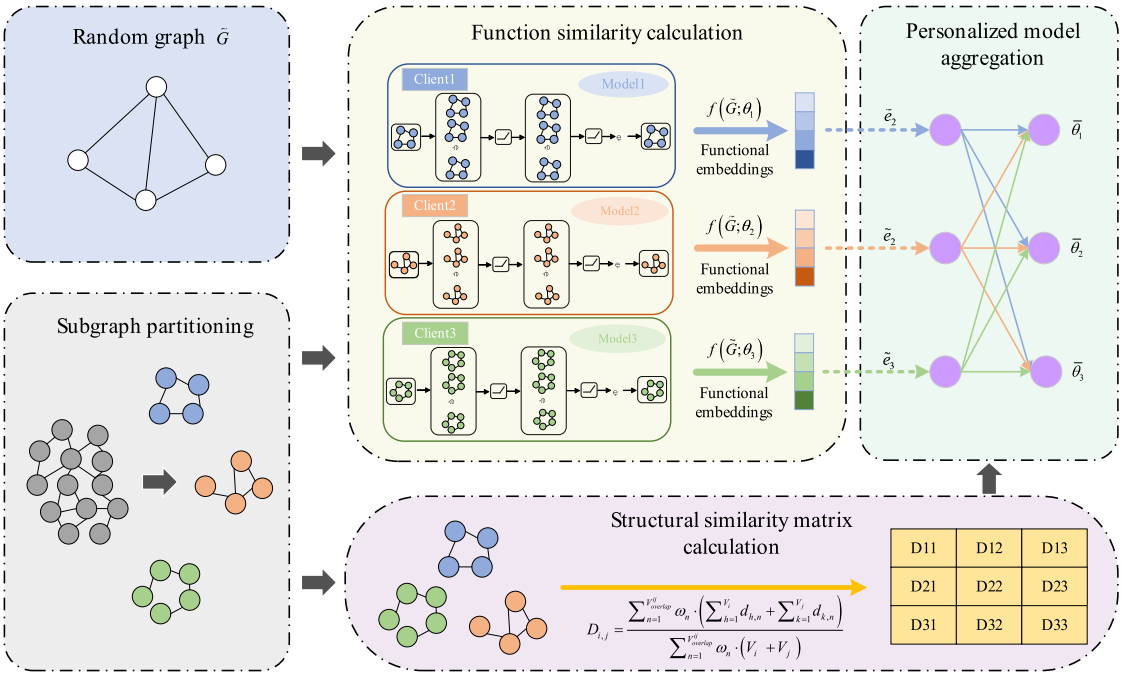


Figure 2: An overview of our proposed SubPFed.

Table 2: Representation and definition of symbols.

Symbol	Definition
M	The total number of clients
$\mathcal{S}$	The central server
$\mathcal{C}_i$	The client i
$\mathcal{D}_i$	The subgraph dataset of client i
$V_{overlap}^{ij}$	The set of overlapping nodes between subgraph $\mathcal{G}_i$ and subgraph $\mathcal{G}_j$
$\overline{\theta}$	The universal set of parameters
θ_i	The model parameters of client i
$\tilde{G}$	The random graph
$S(i,j)$	The functional embeddings for subgraph similarity
α_{ij}	The weight aggregation coefficient between client i and client j
f	The forward propagation function of the client-side local model
$\tilde{e}_i$	The functional embedding of client i to random graph
n	The overlapping node
ω_n	The weight calculated based on the degree of overlapping node n

(Continued)

Table 2 (continued)

Symbol	Definition
V_i	The node set of subgraph $\mathcal{G}_i$
$d_{h,n}$	The shortest-path distance from node h to overlapping node n in subgraph $\mathcal{G}_i$
ψ	The weight of structural information
$D_{i,j}$	The structural information for subgraph similarity
λ	The L2 proximal term coefficient
ρ	The overlapping ratio

Next, we will provide a detailed elaboration of the specific workflow of the SubPFed approach.

1. System Initialization. Utilize graph partitioning tools such as Metis to divide the original large graph $\mathcal{G}$ into M subgraphs $\{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_m\}$. Subsequently, assign each client a dedicated subgraph $\mathcal{G}_i$ and initialize the model parameters $\overline{\theta}$.
2. Similarity Calculation. For every pair of clients, extract the set of overlapping nodes $V_{overlap}^{ij}$. Compute the structural similarity $D_{i,j}$ by evaluating the weighted distance based on the degree of overlapping nodes as defined in Formula (5), where the weight is determined by the degree of the overlapping nodes. Additionally, the server generates a random graph $\tilde{G}$ and distributes it to all clients. Each client employs the local GNN model to propagate the random graph forward and generate node embeddings. The functional embedding vector $\tilde{e}_i$ of the client is obtained by averaging all node embeddings. Functional similarity $S(i, j)$ is then calculated using the cosine similarity between the functional embeddings of clients.
3. Model Training and Aggregation

During local training on the client side, introduce L2 regularization as shown in Formula (7) to mitigate overfitting of local data. During server-side aggregation, client parameters are weighted and averaged according to the comprehensive similarity as shown in Formula (6): $\overline{\theta}_i = \sum_{j=1}^{M} \alpha_{ij}\theta_j$, where α_{ij} represents the normalized comprehensive similarity weight and ψ denotes the structural information weight.

4. Personalized FL

Each client obtains a personalized model $\{\theta_1, \theta_2, \ldots, \theta_M\}$ that adapts to the distribution of its local data.

The training and execution processes of the client-side and server-side algorithms proposed in SubPEed are summarized in Algorithms 1 and 2.

Algorithm 1: SubPFed client algorithm

Input: the number of rounds R; the number of epochs E; the number of clients M; local subgraph G_i for the i^{th} client; model function f_i for client i; initial parameters $\overline{\theta}_i$ for client i; learning rate η

Output: updated model parameters θ_i

Function RunClient($\overline{\theta}_i$)

    $\theta_i \leftarrow \overline{\theta}_i$

    for each local epoch e from 1 to E do

(Continued)

Algorithm 1 (continued)

$\theta_i \leftarrow \theta_i - \eta \nabla \mathcal{L}\left(G_i; \theta_i\right)$
end for
return θ_i

Algorithm 2: SubPFed server algorithm

Input: the number of rounds R; the number of clients M; the weight of Structural Information ψ; functional embeddings for subgraph similarity $S(\cdot)$; structural information for subgraph similarity $D_{i,j}$

Output: final model parameters of each client $\theta_i^{(R+1)}$

Function RunServer($\overline{\theta}_i$)
Initialize $\overline{\theta}^{(1)}$
for each round $r = 1, 2, \ldots, R$ do
for $\forall i$ in parallel do
if $r = 1$ then
$\theta_i^{(r+1)} \leftarrow$RunClient($\overline{\theta}^{(r)}$)
else
$\overline{\theta}_i^{(r)} \leftarrow \sum_{j=1}^{M} \left(\frac{\psi}{\sum_M D_{i,m}} \times D_{i,j} + \frac{1-\psi}{\sum_M S(i,m)} \cdot S(i,j) \right) \cdot \theta_j$
$\theta_i^{(r+1)} \leftarrow$RunClient($\overline{\theta}_i^{(r)}$)
end if
end for
end for

4.2 Subgraph Similarity Based on Functional Embedding

Under the framework of FL, calculating the similarity of local subgraphs without directly accessing them poses a significant challenge. A widely adopted approach is to use model parameters as proxies, as GNNs trained on subgraphs embed their structural knowledge into these parameters. Let S denote the similarity measure, and θ represent the flattening of parameters into vectors, as follows: $S(i,j) = \frac{\theta_i \cdot \theta_j}{\|\theta_i\| \|\theta_j\|}$. However, the parameter space of the model is typically high-dimensional. Directly computing the similarity between parameters may be influenced by the "curse of dimensionality", leading to inaccurate similarity measurements [35]. Although these parameters encapsulate the model's knowledge, they do not explicitly reflect the specific structure of the subgraph or the intrinsic characteristics of graph data.

To address this issue, we propose leveraging the outputs of GNNs to calculate functional similarity. Specifically, a random graph $\tilde{G} = \left(\tilde{V}, \tilde{\mathcal{E}}\right)$ is designed as input for each client [14], we use the ER model to generate the topological structure of a random graph. The specific parameters are number of node N and connection probability p. In experiments, we usually set number of node to be comparable to the average number of nodes in the equal-scale graph of the dataset. We set a target average node degree d_avg, and then calculate the connection probability based on the formula $p = \frac{d\_avg}{N-1}$ to ensure that the generated random graph is neither completely sparse nor overly dense. The feature vector of each node is independently and identically sampled from a simple prior distribution. To ensure stability and universality, we default to the standard normal distribution. In the experiment, we set the feature dimension d as a fixed dimension to avoid processing complexity caused by different clients having different local data feature dimensions.

Then the forward propagation function f of the GNN model i is computed. Here, $f\left(\tilde{G}; \theta_i\right)$ accepts the graph $\tilde{G}$ and the model parameters θ_i as inputs, returning the embedded representation of each node in the

graph, i.e., obtaining the vector representation of each node. Subsequently, the embeddings of all nodes in the graph are averaged to derive an overall graph embedding $\tilde{e}_i$, which represents the "holistic representation" of the model i on the input graph $\tilde{G}$. Mathematically, this can be expressed as:

$$\tilde{e}_i = \frac{1}{\left|\tilde{V}\right|} \sum_{v \in \tilde{V}} f\left(\tilde{G}; \theta_i\right)(v), \tag{4}$$

where $\tilde{V}$ denotes the set of nodes in graph $\tilde{G}$, and $\sum_{v \in \tilde{V}} f\left(\tilde{G}; \theta_i\right)(v)$ represents the embedding of node v in the GNN model i.

After acquiring the node embeddings from all clients, the cosine similarity between client i and j the model outputs for the clients is calculated. This process defines the functional embedding similarity between two clients as follows: $S(i, j) = \frac{\tilde{e}_i \cdot \tilde{e}_j}{\|\tilde{e}_i\| \|\tilde{e}_j\|}$. During this procedure, access to model parameters and gradients is avoided, thereby ensuring data privacy while achieving computational efficiency.

4.3 Subgraph Similarity Based on Overlapping Nodes

In addition to using the model gradient or the similarity of the model output to determine the similarity of the client data distribution, we find that different subgraphs have overlapping nodes. Among different subgraphs, there are some common nodes, which are shared by multiple subgraphs and play a connecting role in the overall graph structure. For example, for two subgraphs $\mathcal{G}_i$ and $\mathcal{G}_j$, there is an overlapping node set $V_{overlap}^{ij}$ between them, where $V_{overlap}^{ij} = V_i \cap V_j \neq \varnothing$. These nodes across subgraphs usually contain similar information in different subgraphs. Therefore, knowledge can be propagated among multiple subgraphs, promoting information sharing and learning.

For two subgraphs $\mathcal{G}_i$ and $\mathcal{G}_j$, when a short distance is established between the two subgraphs through overlapping nodes, it can be considered that their data distributions are similar. Therefore, the similarity between these subgraphs can be quantified by calculating the distance between them. For two subgraphs $\mathcal{G}_i$ and $\mathcal{G}_j$, there is a set of overlapping nodes $V_{overlap}^{ij}$ between them, where $V_{overlap}^{ij} = \{n_1, n_2, n_3, \ldots\}$ are the shared nodes of the two subgraphs. For each overlapping node n, the distance from the node to this overlapping node is weighted according to the weight ω_n, and then the weighted distance between the two subgraphs is calculated. The final result is obtained by integrating the weighted distances of all overlapping nodes. We define the distance $D_{i,j}$ between two subgraphs as the weighted average distance from all points in the two subgraphs to the overlapping node:

$$D_{i,j} = \frac{\sum_{n=1}^{V_{overlap}^{ij}} \omega_n \cdot \left(\sum_{h=1}^{V_i} d_{h,n} + \sum_{k=1}^{V_j} d_{k,n}\right)}{\sum_{n=1}^{V_{overlap}^{ij}} \omega_n \cdot \left(V_i + V_j\right)}, \tag{5}$$

where, $V_{overlap}^{ij}$ represents the number of overlapping nodes, ω_n is the weight calculated based on the degree of overlapping node n, $d_{h,n}$ is the shortest distance from node h to overlapping node n in subgraph $\mathcal{G}_i$, and similarly, $d_{k,n}$ is the shortest distance from node k to overlapping node n in subgraph $\mathcal{G}_j$, where V_i and V_j are the number of nodes in subgraphs $\mathcal{G}_i$ and $\mathcal{G}_j$, respectively.

Overlapping nodes themselves are only local information of the graph, and sharing these nodes among multiple clients does not leak much data about other parts of the graph [18]. This approach avoids the direct exchange of complete data or gradients between clients, thereby solving the privacy issue. In addition, the weighted average distance can be pre-calculated before model training and only needs to be calculated once, which can greatly reduce the time overhead of model training. This method is designed to comply with the

privacy and efficiency requirements of FL. Unlike traditional graph matching approaches that require full access to graph structures, it leverages only publicly available overlapping node information as a lightweight proxy metric for distributed difference measurement, tailored for federated aggregation.

4.4 Federated Personalized Subgraph Learning Based on Subgraph Similarity Estimation

The function similarity based on subgraph similarity estimation demonstrates higher efficiency and effectiveness compared to similarity measurement methods that rely on model parameters or gradients. This is because it only depends on the output of the model (i.e., node embeddings) and does not require direct access or comparison of model parameters or gradients. Consequently, this approach avoids direct access or exchange of raw graph data, thereby reducing the risk of privacy leakage and ensuring privacy protection. However, a significant limitation of using functional embeddings as a similarity measure is that it overlooks the structural information inherent in the subgraphs themselves. Subgraph similarity is not solely reflected in model parameters but should also account for factors such as nodes, edges, adjacency relationships, and the topological structure of the graph. Relying solely on functional embeddings to measure subgraph similarity may fail to comprehensively capture the actual structural differences and similarities among subgraphs.

To address these issues, we propose a method that weights both functional similarity and graph data similarity from other clients to derive a final comprehensive similarity. Subsequently, we perform a weighted average of all local model parameters. This approach not only leverages the functional similarity of the GNN model to measure the similarity between different clients but also incorporates the structural similarity of graph data distributions across clients. The formula for calculating comprehensive similarity is as follows:

$$\begin{aligned}
&\overline{\theta}_i = \sum_{j=1}^{M} \alpha_{ij}\theta_j, \\
&\alpha_{ij} = \frac{\psi}{\sum_M D_{i,m}} \times \frac{\sum_{n=1}^{V_{overlap}^{ij}} \omega_n \cdot \left(\sum_{h=1}^{V_i} d_{h,n} + \sum_{k=1}^{V_j} d_{k,n}\right)}{\sum_{n=1}^{V_{overlap}^{ij}} \omega_n \cdot \left(V_i + V_j\right)} + \frac{1-\psi}{\sum_M S(i,m)} \cdot \frac{\tilde{e}_i \cdot \tilde{e}_j}{\|\tilde{e}_i\| \left\|\tilde{e}_j\right\|},
\end{aligned} \tag{6}$$

where α_{ij} represents the normalized similarity between the client i and j.

To prevent overfitting during the training process, we constrain the variation range of the model through L2 regularization. Specifically, for each client i, our objective function includes an L2 loss proximal term, expressed as follows:

$$\min_{\theta_i} \mathcal{L}\left(G_i; \theta_i\right) + \lambda \left\|\theta_i - \overline{\theta}_i\right\|_2^2 \tag{7}$$

where $\mathcal{L}$ denotes the loss function with hyperparameters λ on a specific client i.

This personalized solution enables knowledge sharing among interrelated subgraphs while safeguarding data privacy and obtaining information about missing edges. Furthermore, compared to methods that calculate data similarity using model parameters or update gradients, the weighted distance method based on overlapping node degrees integrates the structural information of subgraphs, providing a more comprehensive measurement of subgraph data distribution characteristics.

4.5 Complexity Analysis

This section conducts a complexity analysis of the core computational steps involved in this method, with a particular focus on the most time-consuming part of calculating the shortest path distance. For a graph containing V nodes and E edges, the time complexity of using the breadth-first search algorithm to

calculate the shortest path from a single source node to all other nodes is $O(V+E)$. Therefore, the total time complexity of the worst-case scenario for the naive method of calculating the shortest path between all node pairs (i.e., performing BFS once with each node as the source point) is $O(V(V+E))$. In the FL scenario of this method, let the subgraph scale held by client k be V_k nodes and E_k edges. Then, the time complexity of the shortest distance calculation of this client's subgraph is $O(V_k(V_k+E_k))$. Since $V_k \ll V$ and $E_k \ll E$, this local computational overhead is acceptable in practice.

5 Experiment

We evaluated the performance of SubPFed in the node classification task on three graph datasets under subgraph scenarios with varying node overlapping rates.

5.1 Experiment Settings

We adopted the experimental setup proposed by Baek et al. [14] and constructed distribution subgraphs by partitioning the graph dataset into a certain number of participants, where each local client owned one subgraph as part of the original graph. Specifically, to construct distributed datasets with different node overlapping rates, we utilized the METIS graph partitioning algorithm developed by Han et al. [34]. Unlike the Louvain algorithm, which does not allow specifying the number of subgraphs, the METIS algorithm enables dividing the graph into a predefined number of subgraphs based on the number of clients in FL. We applied the METIS algorithm to divide the original graph into 1, 2, and 3 disjoint subgraphs for 5 clients, 10 clients, and 15 clients, respectively. Subsequently, from each split subgraph, we randomly sampled a certain proportion r of nodes and their associated edges as an overlapping node set and add them to the base subgraph corresponding to each client, forming a subgraph specific to each client. This procedure generated overlapping subgraphs with varying degrees of overlapping rates. The experiments were conducted on three datasets: Cora [36], CiteSeer [36], and PubMed [36], and their statistics are summarized in Table 3.

Table 3: The table demonstrates the heterogeneity in the number of nodes, edges, classes, and subgraphs of the original graph and its divided subgraphs with a ratio of overlapping nodes of 0.1. Ori represents the original graph, and cli denotes the number of clients.

	Cora				CiteSeer				Pubmed			
	Ori	**5 Cli**	**10 Cli**	**15 Cli**	**Ori**	**5 Cli**	**10 Cli**	**15 Cli**	**Ori**	**5 Cli**	**10 Cli**	**15 Cli**
# Classes	7				6				3			
# Nodes	2485	497	249	166	2120	424	212	141	19,717	3943	1972	1314
# Edges	10,138	1866	891	604	7358	1410	675	432	88,648	16,374	7671	5748
Heterogeneity	N/A	0.590	0.606	0.632	N/A	0.517	0.541	0.554	N/A	0.362	0.392	0.411

In this study, we employed two layers of Graph Convolutional Networks (GCNs) [21] as the base GNN for all models. The selected datasets, Cora, CiteSeer, and PubMed, were relatively small. We performed 100 communication rounds of FL, optimizing with Adam [37]. All clients participated in each round of FL. We measured the accuracy of node classification on the client subgraphs and averaged the performance across all clients. For model based on FL, we calculated the average accuracy over three repeated experiments. All experiments were conducted on a machine equipped with a NVIDIA GeForce RTX 3090 GPU, 1.8 GHz Intel Core i7-8565U CPU and 8 GB RAM, and the specific experimental parameters are summarized in Table 4.

Table 4: Experimental parameters setting.

Parameter	Value
Number of clients M	15
L2 proximal term coefficient λ	0.001
Learning rate η	0.01
Weight of structural information ψ	0.1
Communication rounds R	100
Overlapping ratio ρ	0.1
Number of epochs E	1

The framework design of SubPFed is model-independent, and its similarity calculation and aggregation mechanism are independent of the specific GNN implementation. GCN was chosen for the purpose of benchmark comparison, and this method can be seamlessly extended to other mainstream GNN architectures such as GAT and GraphSAGE. Verifying its robustness on diverse architectures is one of our future tasks.

5.2 Main Results

We performed experiments with a node overlapping rate of 0.1 and a number of clients of 5, 10, 15, respectively. The results were summarized in Table 5. Overall, SubPFed demonstrated superior performance compared to other baselines in almost all cases. Specifically, SubPFed outperformed FedGNN and FedSage+, two classical subgraph FL methods. Although FedGNN and FedSage+ addressed missing edge information by generating pseudo-neighbor nodes to expand local subgraphs, this approach struggled to meet the specific requirements of each client regarding local data distribution, leading to inferior performance relative to personalized FL methods. In contrast to FedGCN, which performed neighbor node retrieval but failed to leverage correlations among subgraphs, SubPFed effectively utilized local structural relationships among subgraphs, thereby enhancing the adaptability of the global model and reducing the risk of sensitive information leakage. While Fed-PUB alleviated heterogeneity issues, it neglected critical graph structural information, limiting its personalization capabilities. FedSG jointly encodes topological and feature information on the client side, but it does not utilize the information of missing edges. Moreover, topology-related parameters and feature-related parameters need to be transmitted separately, which definitely increases the complexity of communication. By comparison, SubPFed fully exploited graph structure correlations among clients, enabling each client to obtain a personalized model that better matched with its data distribution. Through capturing both functional characteristics and topological structures of graphs, SubPFed conducted similarity weighted aggregation on the server side. Consequently, SubPFed consistently exhibited superior performance across nearly all cases, validating the effectiveness of our proposed method.

Table 5: The average of three runs for three datasets with client numbers of {5, 10, 15} under an overlapping rate of 0.1 is reported.

Methods	Cora			CiteSeer			Pubmed		
	5	10	15	5	10	15	5	10	15
Local	0.8079	0.7615	0.7942	0.7088	0.6634	0.6573	0.8385	0.8192	0.8164
FedAvg	0.7837	0.7659	0.7726	0.7151	0.6430	0.6482	0.8347	0.8165	0.8037
FedGNN	0.8138	0.7946	0.8054	0.7962	0.7019	0.7186	0.8471	0.8356	0.8294
FedSage+	0.8059	0.7702	0.7841	0.7336	0.7149	0.7087	0.8351	0.8335	0.8223
FedGCN	0.8387	0.8092	0.8165	0.8029	0.7483	0.7314	0.8525	0.8347	0.8351
Fed-PUB	0.8533	0.8124	0.8252	0.8067	0.7721	0.7315	0.8562	0.8493	0.8424
FedSG	0.8576	0.8101	0.8253	0.8138	0.7924	0.7781	0.8590	0.8616	0.8493
SubPFed	0.8691	0.8357	0.8479	0.8392	0.8134	0.8031	0.8704	0.8682	0.8565

5.3 Ablation Study

To comprehensively analyze the contribution of each component, we designed and conducted ablation studies. The experimental results are presented in Table 6. Compared to FedAvg method, personalized FL using the average shortest path distance calculated based on overlapping nodes improved node classification accuracy by up to 16.91%. This indicates that considering overlapping node information between subgraphs effectively captures correlations among subgraphs, thereby enhancing the model's learning ability. Furthermore, our proposed weighted distance method based on overlapping nodes further improved accuracy in personalized FL by 0.56% to 5.35%. Its significant advantage lies in introducing node weights, assigning higher weights to overlapping nodes with higher degrees, thus enhancing attention to key nodes and achieving more accurate subgraph distance estimation. This weighting mechanism describes the similarity relationship between subgraphs at a finer granularity, improving model accuracy during aggregation. Additionally, compared to models based solely on the graph topology structure of overlapping nodes, our proposed SubPFed model achieved a 9.62% improvement in node classification accuracy on the CiteSeer dataset by integrating local subgraph structure information and graph functional characteristics. This improvement ranged from 4.28% to 26.50% over the FedAvg method.

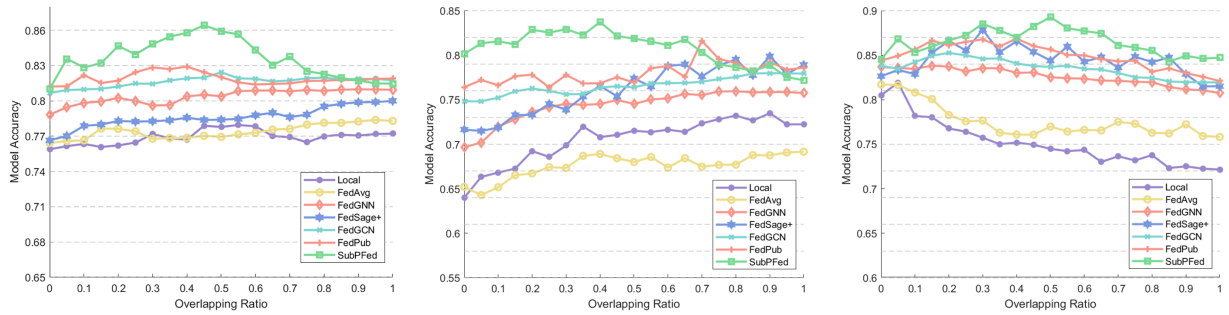
Table 6: Performance in personalized FL with structural information and functional embeddings.

	Cora			CiteSeer			Pubmed		
	5	10	15	5	10	15	5	10	15
FedAvg	0.7837	0.7659	0.7726	0.7151	0.6430	0.6482	0.8347	0.8165	0.8037
The average shortest path distance method	0.8367	0.8146	0.8239	0.7998	0.7517	0.7285	0.8491	0.8422	0.8324
The weighted distance method	0.8456	0.8252	0.8364	0.8169	0.7919	0.7326	0.8548	0.8473	0.8411
SubPFed	0.8691	0.8357	0.8479	0.8392	0.8134	0.8031	0.8704	0.8682	0.8565

Specifically, subgraph structure information captures node connection patterns, while graph functional characteristics provide rich node attribute information. By organically combining these two components, SubPFed offers a more comprehensive description of subgraph characteristics, thereby achieving more accurate personalized FL. Experiments confirmed that each module design significantly contributed to overall performance improvement, validating the rationality and effectiveness of each component in our method. Notably, the SubPFed model that integrates topological structure and functional characteristics demonstrated the most significant advantages in the node classification task.

5.4 The Impact of Node Overlapping Rate

The node overlapping rate is defined as the proportion of overlapping nodes to the total nodes, which significantly influences the estimation of subgraph similarity and the accuracy of server-side aggregation. An appropriate node overlapping rate enables the model to effectively capture correlations among subgraphs, thereby enhancing the performance of node classification tasks. As illustrated in Fig. 3, experimental results indicate that within an overlapping rate range of 0.4 to 0.5, the classification performance of the model surpasses other settings. However, as the proportion of overlapping nodes increases beyond this range, model accuracy gradually diminishes. At lower overlapping rates, insufficient shared nodes among subgraphs lead to inadequate structural information during aggregation, resulting in inaccurate similarity assessments by the model. Conversely, when the proportion of overlapping nodes exceeds 0.7, the performance on the Cora and CiteSeer datasets declines. Although tighter structural information exists between subgraphs, overly similar features during aggregation weaken the distinctiveness of other subgraphs.



(**a**) Cora, client = 10 (**b**) CiteSeer, client = 10 (**c**) Pubmed, client = 10

Figure 3: Fig. 3 (**a**) Represents the accuracy of different models in graph node classification task under 21 different node overlapping rates on the Cora (10 clients). Fig. 3 (**b**) Represents the accuracy of different models in graph node classification task under 21 different node overlapping rates on the CiteSeer (10 clients). Fig. 3 (**c**) Represents the accuracy of different models in graph node classification task under 21 different node overlapping rates on the Pubmed (10 clients).

5.5 Structural Information Weight

In the SubPFed model, both functional characteristics and structural information jointly determine the classification performance of nodes. Functional characteristics correspond to the attribute data of each node, while structural information emphasizes the adjacency relationships between nodes. Within the SubPFed approach, the weight assigned to structural information governs the contribution ratio of functional characteristics and structural information in the node classification task, serving as a crucial hyperparameter influencing model performance. To explore the impact of different structural information weights on model performance, extensive experiments were conducted across three datasets with differing structural information weights.

As depicted in Fig. 4, when weights are relatively small, e.g., between 0.2 and 0.3, the SubPFed model's performance tends to be suboptimal. Conversely, when weights fall within the range of 0.5 to 0.7, the SubPFed model achieves peak performance, particularly on the Cora and CiteSeer datasets, where node classification accuracy markedly improves. This suggests that when weights are excessively low, the model fails to fully leverage the structural information of overlapping nodes in subgraphs, leading to insufficient expression of graph similarity during aggregation and consequently affecting node classification accuracy. When weights are set too high, model performance tends toward saturation or decline, especially on the Pubmed dataset. Given the sparser node relationships in the Pubmed dataset compared to the others, excessively high structural information weights may cause the model to overlook functional characteristics of nodes. Therefore, smaller weight values enable the model to better balance node features and structural information, achieving superior classification accuracy on this dataset.

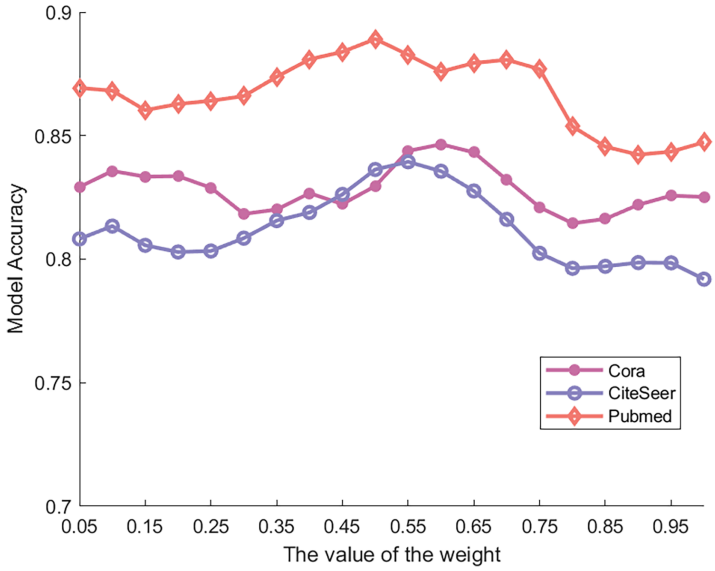


Figure 4: The influence of structural information weight on model accuracy. In order to further explore the influence of structural information weight on the accuracy of the model, the client is set to 10 to conduct experiments on three datasets, respectively, and 20 different structural information weights are set when the overlapping rate is 0.1.

5.6 Handling Missing Edges

To evaluate the SubPFed model's capability in addressing the issue of missing edges, we simulate an environment with missing edge connections by removing subgraph edge links. Existing SFL methods rely on explicitly added edges to establish information flow. In contrast, the SubPFed model employs a random graph as input to compute the function embedding of local GNNs and subsequently derives comprehensive similarity by weighting overlapping nodes with subgraph structural information. Consequently, SubPFed not only addresses the problem of missing edges but also facilitates knowledge sharing through subgraph similarity in the absence of explicit edges. Performance evaluation involves calculating the performance of adjacent subgraphs under missing edge conditions. Since direct edge connections between neighboring subgraphs are absent, we utilize local model weights to evaluate the performance on neighboring subgraphs.

Experimental results, as shown in Fig. 5, SubPFed demonstrates outstanding performance in the environment with missing edges. Compared to relevant baselines, SubPFed exhibits a performance improvement ranging from 3.43% to 7.59%. By implicitly sharing weights among similar subgraphs, SubPFed efficiently shares information without the overhead of explicitly adding edges, showcasing superior performance

across multiple real-world datasets. These findings strongly support SubPFed's effectiveness and potential in handling missing edges within the context of GFL.

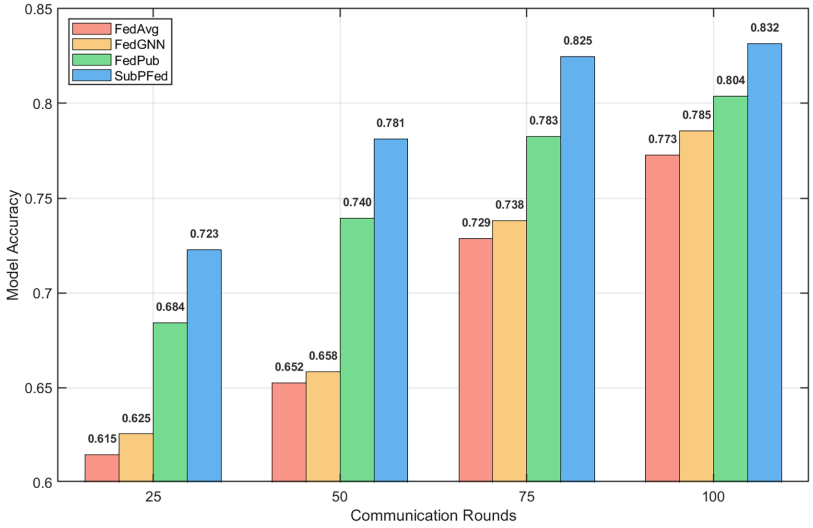


Figure 5: Performances on interrelated neighboring subgraphs of 10 Clients.

5.7 Unbalanced Subgraphs

In real-world scenarios, the sizes of subgraphs often vary significantly. However, in our primary experiment, we assumed that the sizes of subgraphs were relatively similar, which may not align with actual conditions in certain scenarios. To better evaluate the performance of the SubPFed model in an environment with imbalanced subgraphs, we designed an experiment involving different subgraph sizes to simulate a more realistic setting. Initially, the entire graph was divided into 15 subgraphs, each containing a roughly equal number of nodes, ensuring as much balance as possible during the division process. Subsequently, we combined 5, 4, 3, and 2 subgraphs into 4 larger subgraphs, thereby creating an unbalanced subgraph environment. These unbalanced subgraphs were then assigned to 5 clients, and the SubPFed model was trained and evaluated on each client. In this imbalanced subgraph environment, experiments were conducted using the Cora, Citeseer, and Pubmed datasets. In each experiment, the subgraph sizes for the clients differed, and the node distribution across clients was uneven. As shown in Table 7, the SubPFed model demonstrated significant advantages in an unbalanced subgraph environment, particularly in maintaining performance when handling large variations in client data volumes, outperforming relevant baselines.

Table 7: Performance of the three datasets in imbalanced subgraphs.

Model	Cora		CiteSeer		Pubmed	
	5	15	5	15	5	15
Local	0.7961	0.7942	0.6857	0.6573	0.8284	0.8164
FedAvg	0.7789	0.7726	0.6948	0.6482	0.8251	0.8037
FedGNN	0.8115	0.8054	0.7863	0.7186	0.8369	0.8294
FedSage+	0.7923	0.7841	0.7256	0.7087	0.8246	0.8223
FedGCN	0.8247	0.8165	0.7873	0.7314	0.8435	0.8351

(Continued)

Table 7 (continued)

Model	Cora		CiteSeer		Pubmed	
	5	15	5	15	5	15
Fed-PUB	0.8436	0.8252	0.7949	0.7315	0.8492	0.8424
SubPFed	0.8624	0.8479	0.8312	0.8031	0.8694	0.8565

5.8 Communication Efficiency

A key advantage of SubPFed is its high communication and computational efficiency. As shown in Table 8, compared with conventional FedAvg, SubPFed introduces only minimal communication overhead (91.94%) while delivering significant accuracy gains. Notably, existing subgraph FL methods such as FedGNN, FedSage+, FedGCN, and FedSG typically incur 2–5 times higher communication costs than FedAvg. This is mainly due to the exchange of node information between clients for structural estimation, whereas SubPFed avoids explicit information sharing and relies solely on aggregated model updates. Importantly, SubPFed also maintains low memory overhead. As indicated in Table 9 SubPFed operates with a memory footprint of 127 MB, which is lower than FedSage+ (352 MB), FedSG (295 MB), FedGCN (224 MB), and comparable to Fed-PUB (148 MB). This efficiency stems from our server-side pre-computation of structural similarity, which eliminates the need for clients to cache or exchange additional structural data. Combined with its competitive runtime 85.96 min, the lowest among all compared subgraph FL methods SubPFed demonstrates practical feasibility in resource-constrained federated settings where both memory and communication are limited.

Table 8: Analyses on efficiencies of communication costs and time on Pubmed with 10 clients.

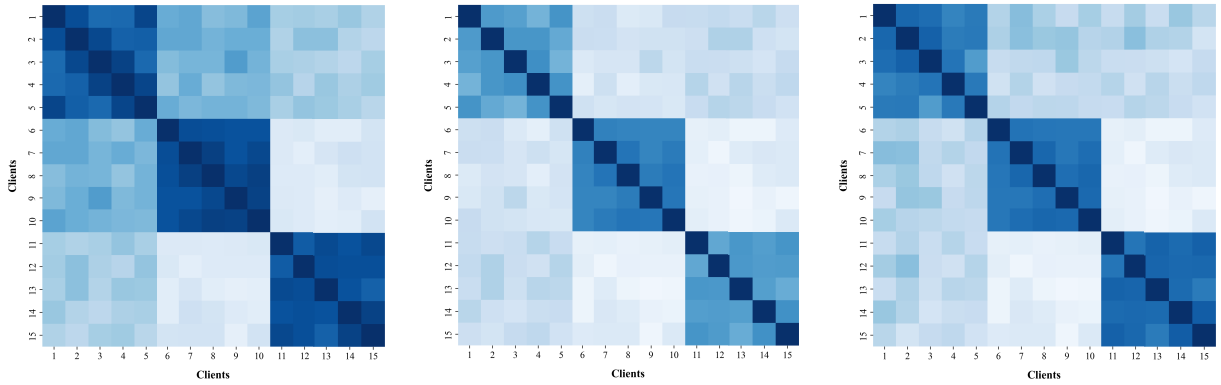
Methods	Accuarcy [%]	Cost [%]	Time [%]
FedAvg	81.65	100	100
FedGNN	83.56	287.05	463.21
FedSage+	83.35	342.54	546.47
FedGCN	83.47	161.79	241.89
Fed-PUB	84.93	90.73	158.95
FedSG	84.51	250.11	227.53
SubPFed	86.82	91.94	130.16

Table 9: Analyses on efficiencies of computational overhead and time on Pubmed with 10 clients.

Methods	Memory [MB]	Time [min]
FedGNN	112	302.42
FedSage+	352	356.65
FedGCN	224	158.07
Fed-PUB	148	103.85
FedSG	295	148.27
SubPFed	127	85.96

5.9 Personalized Aggregation Weight

We aim to prove whether clients with similar data do indeed obtain higher aggregation weights during the process of personalized weight aggregation. For this purpose, based on the similarity information of three types of client structures, we compared and analyzed the distribution of aggregated weights after the 5th and 10th rounds of training. The specific experimental results are shown in Fig. 6. As shown in Fig. 6b, after the 10th round of training, the aggregation weights between clients have initially shown a positive correlation trend with structural similarity. Although the model has not yet fully converged at this point, the aggregation weights of high-similarity clients (such as 1–5) are generally higher than those of low-similarity clients (such as 11–15). This indicates that the personalized aggregation mechanism begins to respond to structural similarity in the early stage of training. By the 50th round of training (Fig. 6c), the distribution of the aggregated weights was further highly aligned with the structural similarity. Specifically, it is manifested that clients 1 to 5 have the highest similarity and obtain significantly higher weights than other clients in the aggregation. The weight values of 6 to 10 on the client side gradually decrease as the similarity decreases, presenting an obvious gradient distribution. The aggregation weights of 11 to 15 on the client side are significantly low. This result indicates that as the number of training rounds increases, the aggregation mechanism continuously intensifies the weight bias towards highly similar clients, ultimately achieving that clients with similar data do indeed obtain higher aggregation weights.



(**a**) Structural similarity (**b**) Round at 10 (**c**) Round at 50

Figure 6: Fig. 6 (**a**) Represents the structural similarity in an overlapping node scenario with Cora (15 clients). Fig. 6 (**b**) Represents the aggregated weights after the 10th round of operation. Fig. 6 (**c**) Represents the aggregated weights after the 50th round of operation.

6 Conclusion

In this study, we propose SubPFed, a novel personalized SFL approach that addresses the challenges of missing edges and data heterogeneity among different clients in SFL. This approach calculates comprehensive similarity between subgraphs by leveraging GNN model function embeddings on random graphs and a weighted distance method based on overlapping node degrees, followed by performing weighted averaging on the local models of each client. We extensively validated our SubPFed approach on multiple benchmark datasets with varying node overlapping rates in subgraphs. On these subgraphs, SubPFed significantly outperformed relevant baselines. Further analysis revealed that our personalized FL approach, based on the weighted distance method of overlapping node degrees, effectively captures structural similarities among subgraphs across different clients. Additionally, the weighted average of the local models of each client, obtained through the fusion of GNN model function embeddings on random graphs and the weighted distance method based on overlapping node degrees, is effective in addressing subgraph heterogeneity. SubPFed

provides a robust solution for personalized SFL and holds broad application potential in cross-domain service collaboration scenarios.

Acknowledgement: Not applicable.

Funding Statement: This work is supported by the Beijing Natural Science Foundation (No. L251061).

Author Contributions: The authors confirm contribution to the paper as follows: conceptualization, Jianbin Li and Hang Bao; methodology, Jianbin Li and Hang Bao; validation, Jianbin Li, Hang Bao and Xin Tong; formal analysis, Jianbin Li and Hang Bao; investigation, Jianbin Li; resources, Hang Bao and Xin Tong; data curation, Hang Bao and Xin Tong; writing—original draft preparation, Jianbin Li and Hang Bao; writing—review and editing, Jianbin Li; visualization, Jianbin Li; supervision, Hang Bao; project administration, Xin Tong; funding acquisition, Jianbin Li. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are openly available in SubPFed at https://github.com/hangbao06/SubPFed.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Hamilton WL. Graph representation learning. In: Synthesis lectures on artificial intelligence and machine learning. 1st ed. San Rafael, CA, USA: Morgan & Claypool; 2020. p. 1–159.
2. Wu Z, Pan S, Chen F, Long G, Zhang C, Yu PS. A comprehensive survey on graph neural networks. IEEE Trans Neural Netw Learn Syst. 2021;32(1):4–24. doi:10.1109/TNNLS.2020.2978386.
3. Zhao T, Zhang X, Wang S. GraphSMOTE: imbalanced node classification on graphs with graph neural networks. In: Proceedings of the 14th ACM International Conference on Web Search and Data Mining; 2021 Mar 8–12; Virtual. p. 833–41.
4. Zhang K, Yang C, Li XX, Sun LC, Yiu SM. Subgraph federated learning with missing neighbor generation. In: Proceedings of the 35th International Conference on Neural Information Processing Systems (NeurIPS); 2021 Dec 6–14; Sydney, Australia. p. 6671–82.
5. Lyu L, Yu H, Zhao J, Yang Q. Threats to federated learning. In: Federated learning. Cham, Switzerland: Springer International Publishing; 2020. p. 3–16. doi:10.1007/978-3-030-63076-8_1.
6. Liu R, Xing P, Deng Z, Li A, Guan C, Yu H. Federated graph neural networks: overview, techniques, and challenges. IEEE Trans Neural Netw Learn Syst. 2025;36(3):4279–95. doi:10.1109/TNNLS.2024.3360429.
7. Tan Y, Liu YX, Long GD, Jiang J, Lu QH, Zhang CQ. Federated learning on non-IID graphs via structural knowledge sharing. In: Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence (AAAI); 2023 Feb 7–14; Washington, DC, USA. p. 9953–61.
8. Huang W, Wan G, Ye M, Du B. Federated graph semantic and structural learning. In: Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence; 2023 Aug 19–25; Macau, China. p. 3830–8.
9. McMahan B, Moore E, Ramage D, Hampson S, Arcas BA. Communication-efficient learning of deep networks from decentralized data. In: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS); 2017 Apr 20–22; Fort Lauderdale, FL, USA. p. 1273–82.
10. Wu CH, Wu FZ, Cao Y, Huang YF, Xie X. FedGNN: federated graph neural network for privacy-preserving recommendation. In: Proceedings of the 38th International Conference on Machine Learning (ICML); 2021 Jul 18–24; Graz, Austria. p. 11291–302.
11. Yao Y, Jin W, Ravi S, Joe-Wong C. FedGCN: convergence-communication tradeoffs in federated training of graph convolutional networks. Adv Neural Inf Process Syst. 2023;36:79748–60. doi:10.52202/075280-3491.

12. Fu X, Zhang B, Dong Y, Chen C, Li J. Federated graph machine learning: a survey of concepts, techniques, and applications. SIGKDD Explor Newsl. 2022;24(2):32–47. doi:10.1145/3575637.3575644.
13. Xie H, Ma J, Xiong L, Yang C. Federated graph classification over non-IID graphs. Adv Neural Inf Process Syst. 2021;34:18839–52.
14. Baek J, Jeong W, Jin JD, Yoon J, Hwang SJ. Personalized subgraph federated learning. In: Proceedings of the 40th International Conference on Machine Learning (ICML); 2023 Jul 23–29; Honolulu, HI, USA. p. 1396–415.
15. Wang Y, Guo S, Qiao D, Liu G, Li M. FedSG: a personalized subgraph federated learning framework on multiple non-IID graphs. IEEE Trans Emerg Top Comput Intell. 2024;8(5):3678–90. doi:10.1109/TETCI.2024.3372381.
16. Deng P, Liu X, Niu J, Hu C. GraphFed: a personalized subgraph federated learning framework for non-IID graphs. In: Proceedings of the 2023 IEEE 20th International Conference on Mobile Ad Hoc and Smart Systems (MASS); 2023 Sep 25–27; Toronto, ON, Canada. p. 227–33.
17. Li T, Sahu AK, Talwalkar A, Smith V. Federated learning: challenges, methods, and future directions. IEEE Signal Process Mag. 2020;37(3):50–60. doi:10.1109/MSP.2020.2975749.
18. Yang CC, Mao WJ, Zheng XL, Wang H. Privacy-preserving social network integration, analysis, and mining. In: Intelligent systems for security informatics. 1st ed. Oxford, UK: Academic Press; 2013. p. 51–67.
19. Bui TD, Ravi S, Ramavajjala V. Neural graph learning: training neural networks using graphs. In: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining; 2018 Feb 5–9; Los Angeles, CA, USA. p. 364–72.
20. Gilmer J, Schoenholz SS, Riley PF, Vinyals O, Dahl GE. Neural message passing for quantum chemistry. In: Proceedings of the 34th International Conference on Machine Learning (ICML); 2017 Aug 6–11; Sydney, Australia. p. 1263–72.
21. Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. In: Proceedings of the 5th International Conference on Learning Representations (ICLR); 2017 Apr 24–26; Toulon, France.
22. Hamilton WL, Ying R, Leskovec J. Inductive representation learning on large graphs. In: Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS); 2017 Dec 4–9; Long Beach, CA, USA. p. 1025–35.
23. Velickovic P, Cucurull G, Casanova A, Romero A, Liò P, Bengio Y. Graph attention networks. arXiv:1710.10903. 2017.
24. Fofanah AJ, Leigh AO. EATSA-GNN: edge-aware and two-stage attention for enhancing graph neural networks based on teacher-student mechanisms for graph node classification. Neurocomputing. 2025;612(3):128686. doi:10.1016/j.neucom.2024.128686.
25. Li Q, Wen Z, Wu Z, Hu S, Wang N, Li Y, et al. A survey on federated learning systems: vision, hype and reality for data privacy and protection. IEEE Trans Knowl Data Eng. 2023;35(4):3347–66. doi:10.1109/TKDE.2021.3124599.
26. Li T, Sahu AK, Zaheer M, Sanjabi M, Talwalkar A, Smith V. Federated optimization in heterogeneous networks. In: Proceedings of Machine Learning and Systems (MLSys); 2020 Mar 14–16; Austin, TX, USA. p. 429–50.
27. Zhao S, Liao T, Fu L, Chen C, Bian J, Zheng Z. Data-free knowledge distillation via generator-free data generation for non-IID federated learning. Neural Netw. 2024;179(1):106627. doi:10.1016/j.neunet.2024.106627.
28. Chen F, Luo RM, Dong Z, Li Z, He X. Federated meta-learning with fast convergence and efficient communication. arXiv:1802.07876. 2018.
29. Zhang JQ, Hua Y, Wang H, Song T, Xue ZG, Ma RH, et al. FedALA: adaptive local aggregation for personalized federated learning. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI); 2023 Feb 7–14; Washington, DC, USA. p. 11237–44.
30. Ceyani E, Xie H, Buyukates B, Yang C, Avestimehr S. FedGrAINS: personalized SubGraph federated learning with adaptive neighbor sampling. In: Proceedings of the 2025 SIAM International Conference on Data Mining (SDM); 2025 May 1–3; Alexandra, VA, USA. p. 598–607.
31. He CY, Ceyani E, Balasubramanian K, Annavaram M, Avestimehr S. SpreadGNN: serverless multi-task federated learning for graph neural networks. arXiv:2106.02743. 2021.
32. Girvan M, Newman MEJ. Community structure in social and biological networks. Proc Natl Acad Sci U S A. 2002;99(12):7821–6. doi:10.1073/pnas.122653799.

33. Radicchi F, Castellano C, Cecconi F, Loreto V, Parisi D. Defining and identifying communities in networks. Proc Natl Acad Sci USA. 2004;101(9):2658–63. doi:10.1073/pnas.0400054101.
34. Han P, Hua H, Wang H, Shang J. A graphic partition method based on nodes learning for energy pipelines network simulation. Energy. 2023;282(1):128179. doi:10.1016/j.energy.2023.128179.
35. Bellman R. Dynamic programming. Science. 1966;153(3731):34–7. doi:10.1126/science.153.3731.34.
36. Sen P, Namata G, Bilgic M, Getoor L, Gallagher B, Eliassi-Rad T. Collective classification in network data. AI Mag. 2008;29(3):93–106. doi:10.1609/aimag.v29i3.2157.
37. Kingma DP, Ba J. Adam: a method for stochastic optimization. arXiv:1412.6980. 2014.