FedTAD: Topology-aware Data-free Knowledge Distillation for Subgraph Federated Learning

Yinlin Zhu[1] , **Xunkai Li**[2] , **Zhengyu Wu**[2] , **Di Wu**[1†] , **Miao Hu**[1] , **Rong-Hua Li**[2]

[1]Sun Yat-sen University, Guangzhou, China

[2]Beijing Institute of Technology, Beijing, China

zhuylin27@mail2.sysu.edu.cn, cs.xunkai.li@gmail.com, Jeremywzy96@outlook.com, {wudi27, humiao}@mail.sysu.edu.cn, lironghuabit@126.com

Abstract

Subgraph federated learning (subgraph-FL) is a new distributed paradigm that facilitates the collaborative training of graph neural networks (GNNs) by multi-client subgraphs. Unfortunately, a significant challenge of subgraph-FL arises from subgraph heterogeneity, which stems from node and topology variation, causing the impaired performance of the global GNN. Despite various studies, they have not yet thoroughly investigated the impact mechanism of subgraph heterogeneity. To this end, we decouple node and topology variation, revealing that they correspond to differences in label distribution and structure homophily. Remarkably, these variations lead to significant differences in the class-wise knowledge reliability of multiple local GNNs, misguiding the model aggregation with varying degrees. Building on this insight, we propose topology-aware data-free knowledge distillation technology (FedTAD), enhancing reliable knowledge transfer from the local model to the global model. Extensive experiments on six public datasets consistently demonstrate the superiority of FedTAD over state-of-the-art baselines.

1 Introduction

Graph neural networks (GNNs) have emerged as a promising machine learning paradigm on structured data, exhibiting remarkable performance across diverse AI applications, such as social recommendation [Guo and Wang, 2020] and financial analysis [Yang *et al.*, 2021]. However, most existing GNNs presuppose centralized data storage, which grants a single user or institution access to the entire graph for training. This vulnerable assumption does not apply to many real-world scenarios, as a domain-specific graph may consist of multiple subgraphs, each constrained to be locally accessed due to competition and privacy concerns. For instance, each financial institution constructs its transaction network without sharing privacy-sensitive data with others [Fu *et al.*, 2022]. Consequently, training powerful GNNs using collective intelligence becomes challenging in such distributed scenarios.

To this end, federated graph learning (FGL), the concept of integrating federated learning (FL) with graphs is proposed. Especially, the instance of FGL on a semi-supervised node classification paradigm is known as subgraph-FL. In each training round of this federated setting, each client utilizes its privately stored subgraph to train the local model independently. Then, these local models are uploaded to a trusted central server and aggregated into a global model, which will be broadcast to local clients for the next training round. In the aforementioned multi-client collaborative training process, FedAvg [McMahan *et al.*, 2017] serves as a simple yet effective method, accomplishing global model aggregation by quantifying the sizes of data samples across multiple clients.

Despite extensive study of subgraph-FL, inherent differences in data collection methods often lead to multi-client data variation [Xie *et al.*, 2021], leading to a decline in the model performance, referred to as the *subgraph heterogeneity* [Baek *et al.*, 2023]. Unlike data heterogeneity in conventional FL, subgraph heterogeneity arises from both nodes and topology, as shown in Fig. 1(a). As a result, the conventional federated optimization strategies fail at addressing subgraph heterogeneity (e.g., FedProx [Li *et al.*, 2020] for Non-iid label distributions) due to the emission of topology. To this end, Fed-PUB [Baek *et al.*, 2023] measures subgraph similarity for personalized aggregation. FedGTA [Li *et al.*, 2023] introduce graph mixed moments for topology-aware aggregation. FedSage+[Zhang *et al.*, 2021] and FedGNN[Wu *et al.*, 2021] aspire to reconstruct potential missing edges among clients, representing an implicit resolution to subgraph heterogeneity by aligning local optimization objectives at the data level. Despite the considerable efforts of subgraph heterogeneity, they still have the following limitations.

L1: Lack of in-depth exploration. The impact mechanisms of subgraph heterogeneity from the independent roles of node and topology variation are ambiguous. **Solution:** we empirically investigate and decouple node and topology variation among clients, revealing that they correspond to the difference in label distributions and class-wise homophily (i.e., for a specific class of nodes, the preference of them and their neighbors with similar attributes). These variations lead to different class-wise knowledge reliability. Specifically, for a specific label, the local model trained with a local subgraph that has numerous nodes with this label and exhibits strong class-wise homophily has a reliable prediction.

†Corresponding author.

L2: Sub-optimal model aggregation in subgraph-FL. Due to the local training process encoding patterns of local subgraph into parameters of the local model, class-wise knowledge with different reliability is implicitly contained in these parameters. As a result, the server-side model aggregation is inevitably misled by unreliable class-wise knowledge. **Solution:** we propose FedTAD, as a topology-aware data-free knowledge distillation strategy, which fully considers the differences in the reliability of class-wise knowledge among local GNNs to improve subgraph-FL. Specifically, on the *Client side*, FedTAD first utilizes topology-aware node embeddings to measure the reliability of class-wise knowledge. Then, on the *Server side*, FedTAD employs a generator to model the input space and generates a pseudo graph for transferring reliable knowledge from the multi-client local model to the global model. Remarkably, FedTAD can be viewed as a hot-plugging strategy for any FL optimization strategy, aiming to correct the global model misled by unreliable knowledge during multi-client model aggregation.

Our contributions. (1) *New Observation.* To the best of our knowledge, this paper is the first to investigate the subgraph heterogeneity by decoupling multi-client node and topology variation, providing valuable empirical analysis. (2) *New Method.* We propose FedTAD, which can serve as a hot-plugging post-processor for existing federated optimization strategies, achieving reliable knowledge transfer and alleviating the negative impact of unreliable class-wise knowledge caused by subgraph heterogeneity. (3) *SOTA Performance.* Extensive experiments on six datasets demonstrate that FedTAD outperforms existing baselines (up to 5.3% higher), and improves various state-of-the-art FGL optimization strategies (up to 4.9% performance boost).

2 Preliminaries

Graph Neural Networks. Consider an undirected graph $G = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V}$ as the node-set and $\mathcal{E}$ as the edge-set. Each node $v_i \in \mathcal{V}$ has a F-dimensional attribute vector $\boldsymbol{x}_i$ and a label y_i. The adjacency matrix (including self-loops) is $\hat{\mathbf{A}} \in \mathbb{R}^{N \times N}$. $\mathcal{N}(v_i)$ represents the neighbor nodes set of node v_i. Building upon this, most GNNs can be subsumed into the deep message-passing framework [Gilmer *et al.*, 2017], which obtains a representation of the current node by recursively aggregating and transforming the representations of its neighbors. Specifically, the representation of node v_i at the l-th layer is denoted as h_i^{l+1} and is computed as follows:

$$\boldsymbol{h}_i^{l+1} = \mathrm{UPD}(\boldsymbol{h}_i^l, \mathrm{AGG}(\{\boldsymbol{h}_j^l : v_j \in \mathcal{N}(v_i)\})), \qquad (1)$$

where $\boldsymbol{h}_i^0 = \boldsymbol{x}_i$, $\boldsymbol{h}_i^l$ is the representation of node v_i in the l-th (previous) layer, $\mathrm{AGG}(\cdot)$ aggregates the neighbor representations, and $\mathrm{UPD}(\cdot,\cdot)$ updates the representation of the current node using its representation and the aggregated neighbor representation at the previous layer.

Subgraph Federated Learning. In subgraph-FL, the k-th client has an undirected subgraph $G_k = (\mathcal{V}_k, \mathcal{E}_k)$ of an implicit global graph $G_g = (\mathcal{V}_g, \mathcal{E}_g)$ (i.e., $\mathcal{V}_k \subseteq \mathcal{V}_g, \mathcal{E}_k \subseteq \mathcal{E}_g$). Each node $v_i \in \mathcal{V}_k$ has an attribute vector $\boldsymbol{x}_k^i$ and a label y_k^i. Typically, the training process of the t-th communication round in subgraph-FL with FedAvg aggregation strategy can be described as follows: (i) *Client Selection*: Randomly select a set of clients S_t based on a certain probability for this round of subgraph-FL. (ii) *Local Update*: Each selected client (e.g., the k-th) downloads the parameters of the global GNN $\widetilde{\boldsymbol{w}}^t$ and performs local training on its private subgraph G_k to minimize $\mathcal{L}(G_k)$, where $\mathcal{L}(G_k)$ is the task loss for semi-supervised node classification on subgraph G_k. (iii) *Global Aggregation*: The server collects the parameters of the local GNNs $\{w_k\}_{k \in S_t}$ to perform parameter aggregation concerning the number of training instances, i.e., $\widetilde{\boldsymbol{w}}^t = \sum_{k \in S_t} \frac{|\mathcal{V}_k|}{N} \boldsymbol{w}_k^t$, where N denotes the total number of nodes for all selected clients.

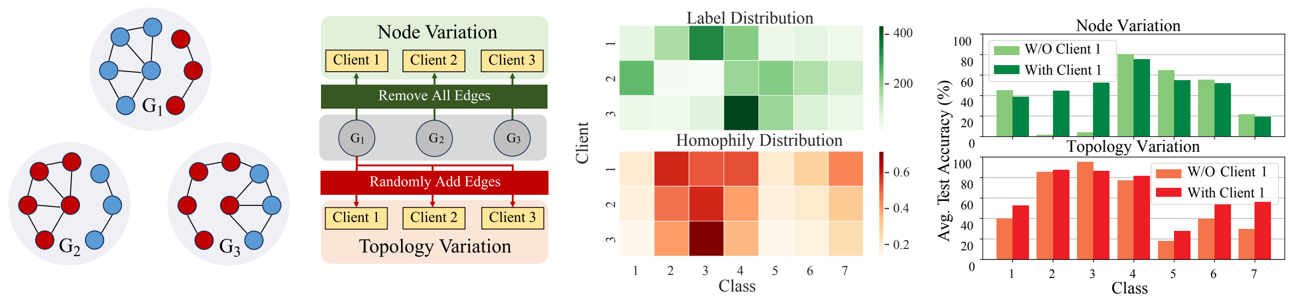
Class-wise Homophily. Various GNN research [Zhu *et al.*, 2020; Sun *et al.*, 2023] has discussed the graph homophily (i.e., for an entire graph, the preference of connected nodes with similar attributes). However, this paper focuses on a more fine-grained issue: class-wise graph homophily (i.e., for a specific class of nodes, the preference of them and their neighbors with similar attributes). We contend that local GNNs offer more reliable predictions for classes with higher class-level homophily, providing effective guidance for subgraph-FL. Details are presented in Section 3.

3 Empirical Investigation

In this section, we delve into subgraph heterogeneity by separating node and topology entanglement. We first apply the average assignment-based Louvain algorithm [Blondel *et al.*, 2008] to split the Cora dataset [Yang *et al.*, 2016] into three subgraphs, aligning with the strategy in FedSage+ [Zhang *et al.*, 2021]. Before being distributed to clients, these subgraphs are modified to decouple the node and topology variation, which are detailed in Fig. 1(b). (i) *Node variation.* Removing edges to mitigate topology variation effects. (ii) *Topology variation.* Clients share the nodes of subgraph 1, eliminating node variation, while the topology is perturbed by randomly adding edges. Then, we conduct subgraph-FL using GCN and FedAvg, comparing global GNN performance with/without Client 1 participation.

Node Variation. Essentially, node variation can be regarded as the data heterogeneity in conventional FL, manifested as label Non-independent identical distribution (Non-iid). Node heterogeneity and the performance of two global models (with/without Client 1 participation) are depicted in Fig. 1(c) and (d). Notably, client 1 predominantly collects nodes of classes 2 and 3, while other class nodes are scarce on Client 1. Consequently, global model performance is notably improved for classes 2 and 3 with Client 1 participation, but it declines to some extent for other classes. Thus, we argue that the node variation leads to differences in the reliability of class-wise knowledge. **Observation 1.** For a specific class, local GNN trained with a larger number of nodes corresponds to more reliable knowledge.

Topology Variation. Building on the previous exploration, where node variation influences class-level knowledge reliability within local GNNs, we assume that topology variation



(a) Subgraph Heterogeneity (b) Data Simulation (c) Label and Homophily Distributions (d) Performance Comparisons

Figure 1: (a) Illustration of subgraph heterogeneity. G_1 and G_2 exhibit different node label distributions (i.e., node variation); G_2 and G_3 have the same node label distributions but significant differences in topological properties (i.e., topology variation). (b) Two data simulation methods of our empirical study. (c) Upper: label distribution under the node variation scenario, the deeper color corresponds to a larger number of nodes; Lower: class-wise homophily distribution under the topology variation scenario, the deeper color corresponds to a stronger class-wise homophily. (d) Performance of global models with/without Client 1 participation under the node variation scenario (upper) and the topology variation scenario (lower).

may yield similar effects. Yet, a critical challenge is properly mapping the intricate topology into class-level distribution. Referring to a previous study [Zhu *et al.*, 2020], GNNs generally exhibit improved performance when dealing with graphs characterized by higher homophily. Intuitively, we assume that GNNs offer more reliable class-wise knowledge when the majority of nodes within that class exhibit stronger local homophily. Consequently, we adapt the *edge homophily ratio* [Zhu *et al.*, 2020] to quantify class-wise homophily. Specifically, consider a graph $G = (\mathcal{V}, \mathcal{E})$ with node label y_i for each node $v_i \in \mathcal{V}$, we use $\varphi(c)$ to denotes the class-wise homophily on class c, which is computed as:

$$\varphi(c) = \frac{|\{(v_p, v_q) : (v_p, v_q) \in \mathcal{E} \wedge (y_p = c \wedge y_q = c)\}|}{|\{(v_p, v_q) : (v_p, v_q) \in \mathcal{E} \wedge (y_p = c \vee y_q = c)\}|}. \quad (2)$$

Fig. 1(c) and (d) display the class-wise homophily distribution of three clients and the performance of two global models (with/without Client 1 participation). Notably, Client 1 exhibits weak class-wise homophily in class 3 but strong homophily in other classes. Global model performance for class 3 declines with Client 1 participation but improves for other classes. We assert that topology variation corresponds to differences in class-wise homophily and also leads to variations in the reliability of class-wise knowledge. **Observation 2.** For a specific class, local GNN trained with stronger class-wise homophily corresponds to more reliable knowledge.

4 Methodology

4.1 Overview

The overview of our proposed FedTAD is depicted in Fig.2. On the *Client side*, each client computes topology-aware node embeddings to measure class-wise knowledge reliability, which is then uploaded to the server. On the *Server side*, guided by the class-wise knowledge reliability, FedTAD generates a pseudo graph for transferring reliable knowledge from multi-client local models to the global model.

4.2 Class-wise Knowledge Reliability Measuring

To obtain reliable knowledge from clients, our key insight is that the reliability of local models for specific label predictions depends on their local subgraph (i) a large number of corresponding labeled nodes and (ii) strong class-wise graph homophily. However, the existence of unlabeled nodes hinders access to class-wise graph homophily. To this end, we introduce a topology-aware node embedding, which characterizes a node with its original feature and local topological structure. Afterward, we assess the class-wise knowledge reliability by quantifying the similarity in topology-aware node embeddings between labeled nodes and their neighbors.

Topology-aware Node Embedding. We obtain the topology embedding of each labeled node i based on the graph diffusion [Tong *et al.*, 2006], which is defined as:

$$\boldsymbol{h}_i^{\text{TOPO}} = [\boldsymbol{T}_{ii}^1, \boldsymbol{T}_{ii}^2, ..., \boldsymbol{T}_{ii}^p] \in \mathbb{R}^p, \quad (3)$$

where $\boldsymbol{T} = \boldsymbol{A}\boldsymbol{D}^{-1}$ represents the random walk transition matrix, $\boldsymbol{D}$ represents the degree matrix (i.e., $\boldsymbol{d}_i = \sum_{j=1}^n \boldsymbol{a}_{ij}$). By combining different walking distances $\{1, ..., p\}$, the topology embedding captures rich structural information. Then, the topology-aware node embedding of v_i can be obtained by concatenating the node features $\boldsymbol{x}_i$ and $\boldsymbol{h}_i^{\text{TOPO}}$:

$$\boldsymbol{h}_i^{\text{HYB}} = \boldsymbol{x}_i || \boldsymbol{h}_i^{\text{TOPO}}. \quad (4)$$

Class-wise Knowledge Reliability. Building upon the above conceptions, for a specific class c, the class-wise knowledge reliability is formally defined as:

$$\phi_c = \sum_{v_i \in \mathcal{V}^{\text{labeled}}} \frac{\sum_{v_j \in \mathcal{N}(v_i)} \boldsymbol{s}(\boldsymbol{h}_i^{\text{HYB}}, \boldsymbol{h}_j^{\text{HYB}})}{|\mathcal{N}(v_i)|}, \quad (5)$$

where $\boldsymbol{s}(\cdot, \cdot)$ represents the cosine similarity. Intuitively, Eq. (5) considers both the label and structure distribution, as the $\mathcal{V}^{\text{labeled}}$ term performs the summation of labeled nodes of the same class, and the $\sum_{v_j \in \mathcal{N}(v_i)} \boldsymbol{s}(\boldsymbol{h}_i^{\text{HYB}}, \boldsymbol{h}_j^{\text{HYB}}) / |\mathcal{N}(v_i)|$ term measures the class-wise graph homophily within the 1-hop ego-graph of each labeled node.

4.3 Topology-aware Knowledge Distillation

The proposed FedTAD servers as a post-processor following the vanilla model aggregators (e.g., FedAvg). Its objective is to transfer reliable knowledge from multi-client local

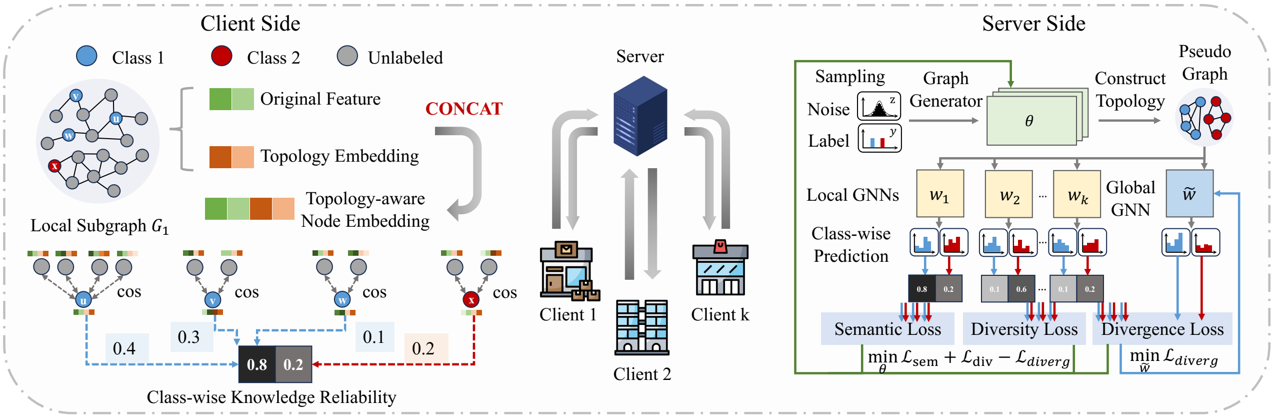


Figure 2: An overview of our proposed FedTAD framework. On the client side, each client performs local initialization for measuring class-wise knowledge reliability. On the server side, the FedTAD can be regarded as a post-process of vanilla model aggregation, which enhances reliable class-wise knowledge transferring from local models to the global model.

models to further improve the aggregated model $\widetilde{\boldsymbol{w}}$. Specifically, in each round t, the server receives the local model weights $\{\boldsymbol{w}_k\}_{k\in S_t}$ and the class-wise knowledge reliability $\{\boldsymbol{\phi}_k\}_{k\in S_t}$ uploaded from the selected clients S_t.

Server-side Knowledge Generation. The server first uses a generator to generate pseudo node attributes $\hat{\boldsymbol{X}} \in \mathbb{R}^{B\times F}$,

$$\hat{\boldsymbol{X}} = P(\boldsymbol{z}, \boldsymbol{y}; \boldsymbol{\theta}), \tag{6}$$

where P denotes the generator, $\boldsymbol{\theta}$ is the parameter of generator, $\boldsymbol{z}$ is the standard Gaussian noise, i.e. $\boldsymbol{z} \sim \boldsymbol{\mathcal{N}}(\boldsymbol{0}, \boldsymbol{1})$, y denotes the label of $\hat{\boldsymbol{x}}$ sampled from uniform distribution, and B denotes the number of generated nodes.

Subsequently, we construct the topology structure based on $\hat{\boldsymbol{X}}$ with the *K-Nearest Neighbors* strategy. Specifically, the pseudo adjacency matrix $\hat{\boldsymbol{A}}$ is computed as follows:

$$\boldsymbol{H} = \sigma(\hat{\boldsymbol{X}}\hat{\boldsymbol{X}}^T),\ \hat{\boldsymbol{A}}[u,v] = \begin{cases} 1, \text{if } v \in \text{TopK}(\boldsymbol{H}[u,:]); \\ 0, \text{otherwise}, \end{cases} \tag{7}$$

where $\sigma(\cdot)$ represents the sigmoid function. The obtained pseudo graph is denoted as $\hat{G} = (\hat{\mathcal{V}}, \hat{\mathcal{E}})$ with the pseudo features $\hat{\boldsymbol{X}}$ and pseudo labels $\hat{\boldsymbol{Y}}$.

To ensure that generated samples closely resemble real ones, it is crucial to confine them within the same semantic space. In many data-free knowledge distillation studies in the vision domain, a strategy involves training the generator to be accurately predicted by teacher models [Chen *et al.*, 2019]. Building upon this approach, we further encourage the generator to learn reliable class-wise semantics from each local model by minimizing the following loss:

$$\mathcal{L}_{sem} = \sum_{k\in S_t}^{K}\sum_{c=1}^{C}\frac{\phi_k^c}{\phi^c}\sum_{v_i\in\hat{\mathcal{V}}_c}\mathcal{L}_{\text{CE}}(\sigma(f(v_i,\hat{G};\boldsymbol{w}_k)), \hat{\boldsymbol{y}}_i), \tag{8}$$

where $\phi^c = \sum_{k\in S_t}\phi_k^c$, $\hat{\mathcal{V}}_c$ denotes the pseudo node with label c, $\sigma(\cdot)$ is the softmax function, f is the GNN classifier, and $\mathcal{L}_{\text{CE}}$ represents the cross-entropy loss. Eq. (8) dynamically adjusts the weights based on the global proportion of class-wise knowledge reliability, ensuring that the generated nodes can be precisely predicted by reliable clients.

Furthermore, to mitigate the risk of mode collapse, the generator is encouraged to minimize diversity loss, facilitating the generation of diverse nodes, which is widely applied in generative models [Ding *et al.*, 2018; Mao *et al.*, 2019].

$$\mathcal{L}_{div} = \frac{\sum_{i,j\in\{1,\dots,B\}}}{B^2}\frac{\hat{\boldsymbol{x}}_i^T\hat{\boldsymbol{x}}_j}{\|\hat{\boldsymbol{x}}_i\|_2\|\hat{\boldsymbol{x}}_j\|_2}. \tag{9}$$

Eq. (9) can reduce the similarity of features across pairs of pseudo nodes, aiming to mitigate redundant features to prevent inefficiencies in subsequent knowledge distillation.

Server-side Knowledge Distillation. The pseudo graph is subsequently employed for knowledge distillation. Owing to the global model being misled by unreliable knowledge, predictions of the global and local models exhibit notable differences in specific nodes. Consequently, for a specific local model, we incentivize the global model to minimize divergence on pseudo-nodes in reliable classes,

$$\mathcal{L}_{diverg} = \sum_{k\in S_t}^{K}\sum_{c=1}^{C}\frac{\phi_k^c}{\phi^c}\sum_{v_i\in\hat{\mathcal{V}}_c}\text{KL}(f(v_i,\hat{G};\widetilde{\boldsymbol{w}})||f(v_i,\hat{G};\boldsymbol{w}_k)), \tag{10}$$

where $\text{KL}(\cdot||\cdot)$ denotes the Kullback-Leibler divergence function. Conversely, the generator aims to generate pseudo nodes that induce divergence between the global and local model in reliable classes as much as possible. Therefore, the generation stage and the distillation stage compose an adversarial training process, which is formulated as follows:

$$\min_{\widetilde{\boldsymbol{w}}}\max_{\boldsymbol{\theta}}\mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{1})}[\mathcal{L}_{diverg} - \lambda_1\mathcal{L}_{sem} - \lambda_2\mathcal{L}_{div}], \tag{11}$$

where λ_1 and λ_2 are trade-off parameters to control the contribution of different loss functions. The overall process is topology-aware since it is guided by class-wise knowledge reliability, considering class-wise graph homophily. The complete algorithm of FedTAD is presented in Algorithm 1.

5 Experiments

In this section, we conduct experiments to verify the effectiveness of FedTAD. Specifically, we aim to answer the following questions: **Q1**: Compared with other state-of-the-art

Algorithm 1 FedTAD Execution on Client and Server

Input:
Rounds, T; Local subgraphs, $\{G_k\}_{k \in S_t}$; Knowledge reliabilities, $\{\phi_k\}_{k \in S_t}$; Server-side training iteration, I; generation iteration, I_g; distillation iteration, I_d;
Output:
Global GNN weight, $\widetilde{\boldsymbol{w}}$;
<u>*Client-side Initialize*</u>:
Obtain topology-aware node embeddings via Eqs. (3, 4);
Obtain the class-wise knowledge reliability via Eq. (5).
for each communication round $t = 1, ..., T$ **do**
  <u>*Client-side Execute*</u>:
  Update local GNN $\{\boldsymbol{w}_k\}_{k \in S_t}$ with $\widetilde{\boldsymbol{w}}$;
  Train local GNN $\{\boldsymbol{w}_k\}_{k \in S_t}$ with $\{G_k\}_{k \in S_t}$;
  <u>*Server-side Execute*</u>:
  Aggregate local GNN $\{\boldsymbol{w}_k\}_{k \in S_t}$ into global GNN $\widetilde{\boldsymbol{w}}$;
  for each iteration $i = 1, ..., I$ **do**
    Generate pseudo node attributes $\hat{\boldsymbol{X}}$ via Eq. (6);
    Obtain the pseudo adjacency $\hat{\boldsymbol{A}}$ via Eq. (7);
    for each inner iteration $g = 1, ..., I_g$ **do**
      Update the generator $\boldsymbol{\theta}$ via Eq. (11);
    end for
    for each inner iteration $d = 1, ..., I_d$ **do**
      Update the global model $\widetilde{\boldsymbol{w}}$ via Eq. (11);
    end for
  end for
end for

federated optimization strategies, can FedTAD achieve better performance? **Q2**: Where does the performance gain of FedTAD come from? **Q3**: Is FedTAD sensitive to the hyperparameters? **Q4**: Can FedTAD maintain robustness when partial clients participate? **Q5**: As a hot-plug method, how much performance improvement can FedTAD bring? **Q6**: What is the time complexity of FedTAD?

5.1 Datasets and Simulation Method

We perform experiments on six widely used public benchmark datasets in graph learning: three small-scale citation network datasets (Cora, CiteSeer, PubMed [Yang *et al.*, 2016]), two medium-scale co-author datasets (CS, Physics [Shchur *et al.*, 2018]), and one large-scale OGB dataset (ogbn-arxiv [Hu *et al.*, 2020]). More details can be found in Table 1. To simulate the distributed subgraphs in the subgraph-FL, we employ the Louvain algorithm [Blondel *et al.*, 2008] to achieve graph partitioning, which is based on the optimization of modularity and widely used in subgraph-FL fields [Zhang *et al.*, 2021; He *et al.*, 2021].

5.2 Baselines and Experimental Settings

Baselines. We compare the proposed FedTAD with five conventional FL optimization strategies (FedAvg [McMahan *et al.*, 2017], FedProx [Li *et al.*, 2020], SCAFFOLD [Karimireddy *et al.*, 2020], MOON [Li *et al.*, 2021], FedDC [Gao *et al.*, 2022]), two personalized subgraph-FL optimization strategies (Fed-PUB [Baek *et al.*, 2023], FedGTA [Li *et al.*, 2023]), one personalized graph-FL optimization strategy (GCFL+ [Xie *et al.*, 2021]), and one subgraph-FL framework (FedSage+ [Zhang *et al.*, 2021]).

Dataset	#Nodes	#Features	#Edges	#Classes
Cora	2,708	1,433	5,429	7
CiteSeer	3,327	3,703	4,732	6
PubMed	19,717	500	44,338	3
CS	18,333	6,805	81,894	15
Physics	34,493	8,415	247,692	5
ogbn-arxiv	169,343	128	2,315,598	40

Table 1: Statistics of the six public benchmark graph datasets.

Hyperparameters. For each client and the central server, we employ a two-layer GCN as our backbone. The dimension of the hidden layer is set to 64 or 128. The local training epoch and round are set to 3 and 100, respectively. The learning rate of GNN is set to 1e-2, the weight decay is set to 5e-4, and the dropout is set to 0.5. Based on this, we perform the hyperparameter search for FedTAD using the Optuna framework [Akiba *et al.*, 2019] on λ_1 and λ_2 within $\{10^{-1}, 10^{-2}, 10^{-3}\}$, and I, I_g, I_d within $\{1, 3, 5, 10\}$. For each experiment, we report the mean and variance results of 3 standardized training.

5.3 Experimental Analysis

Result 1: the answer to Q1. The comparison results are presented in Table 2. As observed, the proposed FedTAD consistently outperforms baselines. Specifically, compared with FedAvg, FedTAD brings at most 5.3% performance improvement; Compared with FedGTA, which is the optimal baseline in most cases, FedTAD can achieve at most 4.9% performance improvement. The convergence curves of FedTAD and baselines are shown in Fig. 3. As observed, FedTAD consistently converges within fewer communication rounds, demonstrating that FedTAD is suitable for subgraph-FL scenarios with limited communication overhead.

Moreover, to reduce privacy leaks, we also evaluated the performance of FedTAD when 10% Gaussian noise is introduced into the uploaded class-level knowledge reliability. As observed, FedTAD exhibits a performance decline of less than 0.4% and remains superior to the baseline.

Result 2: the answer to Q2. To answer **Q2**, we focus on the composition of FedTAD, including the class-wise knowledge reliability module on the client side, and the topology-aware knowledge distillation module on the server side.

First, we conducted an ablation study on the Cora datasets with 5/10/20 participating clients to demonstrate the effectiveness of class-level knowledge reliability (Eq. 5). Specifically, we performed two replacements for its calculation. (i) $\phi_c = 1$ (each class is **eq**ually reliable, **FedTAD-eq** for short); (ii) $\phi_c = |\mathcal{V}^{\text{labeled}}|$ (only considering **n**ode **v**ariation while ignoring topology variation, **FedTAD-nv** for short). Moreover, we additionally replace the pseudo graph in FedTAD-eq with the real global graph (**FedTAD-r-eq** for short). The comparative results are presented in Fig. 4 (a). As observed, the performance achieved by FedTAD-eq, FedTAD-r-eq, and FedAvg are nearly identical, demonstrating that blindly conducting knowledge distillation without considering the class-wise knowledge reliability cannot improve the global model,

Dataset (→)	Cora			CiteSeer			PubMed		
Method (↓)	5 Clients	10 Clients	20 Clients	5 Clients	10 Clients	20 Clients	5 Clients	10 Clients	20 Clients
FedAvg	80.6±0.3	73.6±0.4	56.0±0.3	71.5±0.3	68.9±0.2	66.3±0.4	85.6±0.3	82.9±0.0	80.6±0.3
FedProx	80.9±0.2	73.3±0.3	56.3±0.4	71.3±0.2	69.1±0.3	65.9±0.4	85.3±0.8	82.7±0.1	80.5±0.4
SCAFFOLD	81.1±0.2	73.2±0.2	56.5±0.5	72.3±0.2	69.4±0.3	66.5±0.7	85.8±0.4	82.5±0.2	81.1±0.3
MOON	81.5±0.2	73.3±0.4	56.6±0.3	71.8±0.2	68.4±0.3	66.7±0.2	86.1±0.3	82.5±0.3	80.4±0.2
FedDC	81.3±0.4	73.4±0.3	56.3±0.4	71.5±0.4	69.3±0.2	67.1±0.6	85.4±0.2	82.2±0.5	80.8±0.4
GCFL+	81.7±0.3	73.8±0.5	56.6±0.2	72.0±0.3	69.8±0.3	67.5±0.2	85.8±0.3	83.3±0.3	81.3±0.4
Fed-PUB	82.1±0.6	73.8±0.3	56.5±0.5	71.6±0.3	69.3±0.4	67.2±0.2	86.1±0.3	83.1±0.4	81.2±0.3
FedSage+	82.7±0.5	73.9±0.2	58.1±0.7	71.9±0.2	70.2±0.2	67.9±0.1	86.1±0.4	83.8±0.3	81.6±0.7
FedGTA	83.3±0.2	74.1±0.2	58.8±0.2	72.4±0.2	70.5±0.5	68.3±0.3	86.3±0.3	83.7±0.3	82.0±0.4
FedTAD (Ours)	**85.1±0.1**	**75.3±0.4**	**61.3±0.3**	**73.5±0.3**	**71.7±0.4**	**70.2±0.3**	**87.9±0.1**	**84.4±0.4**	**83.5±0.2**
FedTAD (10% Noise)	84.8±0.3	75.1±0.5	60.9±0.3	73.3±0.5	71.4±0.2	69.8±0.2	87.7±0.4	84.3±0.2	83.2±0.3

Dataset (→)	CS			Physics			ogbn-arxiv		
Method (↓)	5 Clients	10 Clients	20 Clients	5 Clients	10 Clients	20 Clients	5 Clients	10 Clients	20 Clients
FedAvg	90.4±0.3	85.9±0.8	83.9±0.3	94.7±0.3	91.6±0.0	90.3±0.5	60.3±0.3	58.4±0.2	55.7±0.3
FedProx	90.5±0.4	86.1±0.7	84.1±0.1	94.7±0.5	91.6±0.2	90.3±0.0	60.2±0.2	58.2±0.2	55.5±0.1
SCAFFOLD	91.0±0.4	85.8±0.2	84.3±0.5	94.9±0.3	91.4±0.3	90.6±0.0	60.4±0.1	58.7±0.5	55.9±0.8
MOON	91.2±0.3	85.7±0.4	84.1±0.7	95.1±0.4	91.3±0.2	91.1±0.3	60.4±0.2	58.6±0.7	55.5±0.4
FedDC	91.4±0.5	86.1±0.3	84.5±0.5	94.6±0.7	91.7±0.4	91.5±0.4	60.9±0.2	58.5±0.4	56.1±0.5
GCFL+	91.9±0.2	86.3±0.1	84.5±0.4	94.9±0.3	92.1±0.1	91.7±0.6	61.5±0.2	58.1±0.1	55.9±0.4
Fed-PUB	90.8±0.4	86.7±0.4	84.7±0.4	94.7±0.3	92.6±0.6	91.2±0.2	61.3±0.1	58.2±0.3	56.1±0.1
FedSage+	91.7±0.5	87.5±0.3	85.4±0.3	94.8±0.4	92.2±0.4	91.8±0.4	62.1±0.3	59.4±0.3	56.3±0.4
FedGTA	92.0±0.2	88.7±0.0	85.2±0.3	95.3±0.4	92.5±0.1	92.1±0.5	62.4±0.3	60.5±0.2	57.3±0.1
FedTAD (Ours)	**94.3±0.4**	**90.2±0.2**	**88.7±0.4**	**96.2±0.2**	**94.1±0.2**	**93.3±0.3**	**63.2±0.1**	**62.0±0.3**	**59.5±0.5**
FedTAD (10% Noise)	94.1±0.3	89.8±0.4	88.3±0.2	96.1±0.4	93.8±0.3	92.9±0.5	62.9±0.3	61.7±0.4	59.1±0.3

Table 2: Performance comparison of test accuracy achieved by FedTAD and baseline models on six datasets. The best results are highlighted in **bold**, suboptimal results are marked with an underline, and the third-best results are indicated by shading.

even when relying on real graph data. Moreover, The performance of FedTAD-nv is consistently lower than FedTAD, suggesting that ignoring topology will lead to suboptimal class-wise knowledge reliability.

Afterward, we conducted an ablation study on the Cora dataset with 5 participating clients to investigate the contribution of each loss in the knowledge distillation module. Specifically, we remove three loss functions ($\mathcal{L}_{sem}$, $\mathcal{L}_{div}$ and $\mathcal{L}_{diverg}$) for the generator. The experimental results are shown in Fig. 4 (b). As observed, removing any of the losses results in performance degradation for FedTAD, indicating that each loss function plays a crucial role in FedTAD.

Result 3: the answer to Q3. To answer **Q3**, we assess the performance of FedTAD with 20 participating clients under diverse combinations of trade-off parameters. Specifically, we tune the trade-off parameters λ_1 and λ_2 within $\{10^{-3}, 10^{-2}, 10^{-1}, 1\}$. The sensitive analysis results are presented in Fig. 5. As observed, the performance fluctuation of FedTAD remains consistently below 1%, demonstrating that FedTAD is insensitive to the value of trade-off parameters.

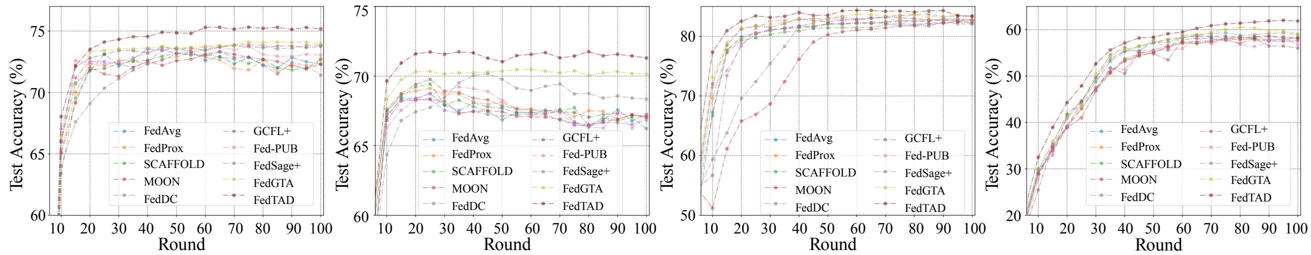
Result 4: the answer to Q4. To answer **Q4**, we compare FedTAD with four baseline methods, including FedAvg, Fed-PUB, FedSage+, and FedGTA on Cora and ogbn-arxiv datasets with 20 participating clients. The activate fraction varies in $\{0.2, 0.4, 0.6, 0.8, 1.0\}$. The experimental results are presented in Table. 4. As observed, FedTAD consistently outperforms the baseline, indicating that FedTAD can maintain robustness with partial client participation.

Result 5: the answer to Q5. To answer **Q5**, we incorporate FedTAD as a plugin into several FGL (i.e., graph-level FL and subgraph-FL) baseline methods, including GCFL+, Fed-PUB, FedSage+, and FedGTA, and evaluate their performance boost on the ogbn-arxiv dataset with 20 participating clients. The experimental results are presented in Table. 3. As observed, FedTAD consistently improves baselines and achieves at most a 5.1% performance boost.

Result 6: the answer to Q6. To answer **Q6**, we provide the complexity analysis of FedTAD. On the Client side, calculating the topology-aware node embedding (Eq. 3) costs $\mathcal{O}(pB)$ with sparse computation, where p denotes the walk distance, and B denotes the number of generated nodes; class-wise knowledge reliability (Eq. 5) costs $\mathcal{O}(m)$, where m denotes the number of edges; On the Server side, the time complexity of model aggregation depends on the chosen optimization strategy; the topology-aware data-free knowledge distillation process costs $\mathcal{O}(I(I_g Bf(z + B + Nc) + I_d BfNc))$, where N denotes the number of participating clients, and z, f, c denote the dimension of sampled noise, node feature, and number of classes, respectively.

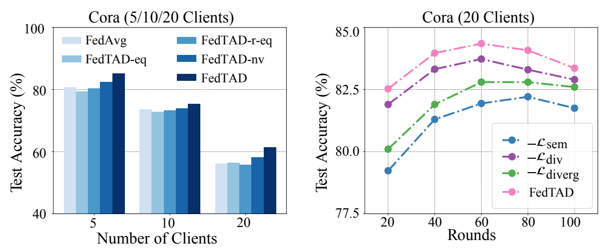
6 Related Work

Graph Neural Networks. Earlier research on deep graph learning extends convolution to handle graphs [Bruna *et al.*, 2013] but comes with notable parameter counts. To this end, GCN [Kipf and Welling, 2016] simplifies graph convolution by utilizing a 1-order Chebyshev filter to capture lo-



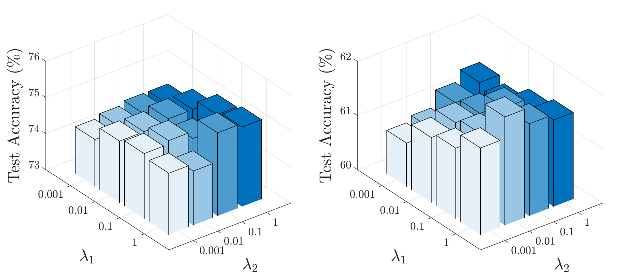
(a) Cora (b) CiteSeer (c) PubMed (d) ogbn-arxiv

Figure 3: Convergence curves of our proposed FedTAD and baseline methods on four graph datasets with 10 participating clients.



(a) CKR Ablation (b) TAD Ablation

Figure 4: Experimental results for the ablation study.



(a) Cora (20 Clients) (b) ogbn-arxiv (20 Clients)

Figure 5: Sensitive analysis for two trade-off parameters λ_1 and λ_2.

Dataset (→)	Cora (20 Clients)		ogbn-arxiv (20 Clients)	
Method (↓)	w/o FedTAD	with FedTAD	w/o FedTAD	with FedTAD
GCFL+	56.6±0.2	61.5±0.1	55.9±0.4	59.4±0.2
Fed-PUB	56.5±0.5	61.6±0.2	56.1±0.1	59.7±0.7
FedSage+	58.1±0.7	61.7±0.4	56.3±0.4	60.1±0.3
FedGTA	58.8±0.2	62.5±0.2	57.3±0.1	60.5±0.4

Table 3: Performance boost on baselines with FedTAD plug.

Dataset (→)	Cora (20 Clients)				
Method (↓)	20% frac.	40% frac.	60% frac.	80% frac.	100% frac.
FedAvg	44.5±0.4	50.3±0.2	51.8±0.3	53.1±0.6	56.0±0.3
Fed-PUB	46.1±0.6	51.9±0.3	52.1±0.5	53.2±0.3	56.5±0.5
FedSage+	47.5±0.5	52.8±0.2	54.4±0.2	56.4±0.3	58.1±0.7
FedGTA	49.6±0.3	53.1±0.3	55.7±0.2	56.2±0.4	58.8±0.2
FedTAD	**51.5±0.2**	**54.2±0.4**	**57.4±0.4**	**58.5±0.4**	**61.3±0.3**

Table 4: Performance of FedTAD on various active fractions.

cal neighborhood information. Moreover, GAT [Veličković *et al.*, 2017] adopts graph attention, allowing weighted aggregation. GraphSAGE [Hamilton *et al.*, 2017] introduces a variety of learnable aggregation functions for performing message aggregation. Further details on GNN research can be found in surveys [Wu *et al.*, 2020; Zhou *et al.*, 2020].

Federated Graph Learning. Mainstream FGL studies can be divided into two settings: (i) Graph-FL: GCFL+ [Xie *et al.*, 2021] and FedStar [Tan *et al.*, 2023]. Each client collects multiple independent graphs, aiming to collaboratively solve graph-level downstream tasks (e.g., graph classification). (ii) Subgraph-FL: each client holds a subgraph of an implicit global graph, aiming to solve node-level downstream tasks (e.g., node classification). Notably, there are currently two main challenges: (a) The global GNN performance degradation caused by *subgraph heterogeneity* (node and topology variation in multi-client subgraphs). Fed-PUB [Baek *et al.*, 2023] points out this heterogeneity stems from the various label distributions and addresses it through personalized technologies. (b) The information loss caused by *missing edges* (connections implicitly exist between subgraphs across clients are lost due to distributed storage). Approaches for handling missing edges include FedSage+ [Zhang *et al.*, 2021] and FedGNN [Wu *et al.*, 2021].

7 Conclusion

In this paper, we provide an in-depth investigation of the subgraph heterogeneity problem by individually analyzing the impact of node and topology variation. We demonstrate that subgraph heterogeneity results in variation in the class-wise knowledge reliability. Building on this insight, we propose FedTAD, utilizing topology-aware data-free knowledge distillation to enable the local models to transfer their most reliable knowledge to correct the global model misled by unreliable knowledge. The experimental results demonstrate that FedTAD significantly outperforms state-of-the-art baselines.

Acknowledgments

This work was supported by the Shenzhen Science and Technology Program (Grant No. KJZD20230923113901004), the Science and Technology Planning Project of Guangdong Province under Grant 2023A0505020006, and the National Natural Science Foundation of China under Grants U2001209, 62072486.

References

[Akiba *et al.*, 2019] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631, 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.

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

[Bruna *et al.*, 2013] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

[Chen *et al.*, 2019] Hanting Chen, Yunhe Wang, Chang Xu, Zhaohui Yang, Chuanjian Liu, Boxin Shi, Chunjing Xu, Chao Xu, and Qi Tian. Data-free learning of student networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3514–3522, 2019.

[Ding *et al.*, 2018] Ming Ding, Jie Tang, and Jie Zhang. Semi-supervised learning on graphs with generative adversarial nets. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 913–922, 2018.

[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.

[Gao *et al.*, 2022] Liang Gao, Huazhu Fu, Li Li, Yingwen Chen, Ming Xu, and Cheng-Zhong Xu. Feddc: Federated learning with non-iid data via local drift decoupling and correction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10112–10121, 2022.

[Gilmer *et al.*, 2017] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.

[Guo and Wang, 2020] Zhiwei Guo and Heng Wang. A deep graph neural network-based mechanism for social recommendations. *IEEE Transactions on Industrial Informatics*, 17(4):2776–2783, 2020.

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

[He *et al.*, 2021] Chaoyang He, Keshav Balasubramanian, Emir Ceyani, Carl Yang, Han Xie, Lichao Sun, Lifang He, Liangwei Yang, Philip S Yu, Yu Rong, et al. Fedgraphnn: A federated learning system and benchmark for graph neural networks. *arXiv preprint arXiv:2104.07145*, 2021.

[Hu *et al.*, 2020] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133, 2020.

[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, 2016] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[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] Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10713–10722, 2021.

[Li *et al.*, 2023] Xunkai Li, Zhengyu Wu, Wentao Zhang, Yinlin Zhu, Ronghua Li, and Guoren Wang. Fedgta: Topology-aware averaging for federated graph learning. *Proc. VLDB Endow.*, 17(1):41–50, 2023.

[Mao *et al.*, 2019] Qi Mao, Hsin-Ying Lee, Hung-Yu Tseng, Siwei Ma, and Ming-Hsuan Yang. Mode seeking generative adversarial networks for diverse image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1429–1437, 2019.

[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.

[Shchur *et al.*, 2018] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868*, 2018.

[Sun *et al.*, 2023] Henan Sun, Xunkai Li, Zhengyu Wu, Daohan Su, Rong-Hua Li, and Guoren Wang. Breaking the entanglement of homophily and heterophily in semi-supervised node classification, 2023.

[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.

[Tong *et al.*, 2006] Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. Fast random walk with restart and its applications. In *Sixth international conference on data mining (ICDM'06)*, pages 613–622. IEEE, 2006.

[Veličković *et al.*, 2017] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[Wu *et al.*, 2020] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.

[Wu *et al.*, 2021] Chuhan Wu, Fangzhao Wu, Yang Cao, Yongfeng Huang, and Xing Xie. Fedgnn: Federated graph neural network for privacy-preserving recommendation. *arXiv preprint arXiv:2102.04925*, 2021.

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

[Yang *et al.*, 2016] Zhilin Yang, William Cohen, and Ruslan Salakhudinov. Revisiting semi-supervised learning with graph embeddings. In *International conference on machine learning*, pages 40–48. PMLR, 2016.

[Yang *et al.*, 2021] Shuo Yang, Zhiqiang Zhang, Jun Zhou, Yang Wang, Wang Sun, Xingyu Zhong, Yanming Fang, Quan Yu, and Yuan Qi. Financial risk analysis for smes with graph-based supply chain mining. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 4661–4667, 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.

[Zhou *et al.*, 2020] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020.

[Zhu *et al.*, 2020] Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. *Advances in neural information processing systems*, 33:7793–7804, 2020.