RESEARCH-ARTICLE

FedGVD: Efficient Federated Graph Learning via Unidirectional Distillation with Dynamic Virtual Nodes

ZHEHAO DAI, Zhejiang University of Technology, Hangzhou, Zhejiang, China

GUOJIANG SHEN, Zhejiang University of Technology, Hangzhou, Zhejiang, China

YUYUE HU, Zhejiang University of Technology, Hangzhou, Zhejiang, China

JIAXIN DU, Zhejiang University of Technology, Hangzhou, Zhejiang, China

XIAO HAN, Zhejiang University of Technology, Hangzhou, Zhejiang, China

XIANGJIE KONG, Zhejiang University of Technology, Hangzhou, Zhejiang, China

Published: 10 November 2025

CIKM '25: The 34th ACM International Conference on Information and Knowledge Management
November 10 - 14, 2025
Seoul, Republic of Korea

Conference Sponsors:
SIGWEB
SIGIR

https://doi.org/10.1145/3746252.3761338
ISBN: 9798400720406

FedGVD: Efficient Federated Graph Learning via Unidirectional Distillation with Dynamic Virtual Nodes

Zhehao Dai
Zhejiang University of Technology, Zhejiang Key Laboratory of Visual Information Intelligent Processing
Hangzhou, China
zhehaodai@outlook.com

Guojiang Shen
Zhejiang University of Technology, Zhejiang Key Laboratory of Visual Information Intelligent Processing
Hangzhou, China
gjshen1975@zjut.edu.cn

Yuyue Hu
Zhejiang University of Technology, Zhejiang Key Laboratory of Visual Information Intelligent Processing
Hangzhou, China
Huyuyue0417@outlook.com

Jiaxin Du
Zhejiang University of Technology, Zhejiang Key Laboratory of Visual Information Intelligent Processing
Hangzhou, China
jiaxin.joyce.du@gmail.com

Xiao Han
Zhejiang University of Technology, Zhejiang Key Laboratory of Visual Information Intelligent Processing
Hangzhou, China
hahahenha@gmail.com

Xiangjie Kong*
Zhejiang University of Technology, Zhejiang Key Laboratory of Visual Information Intelligent Processing
Hangzhou, China
xjkong@ieee.org

Abstract

Federated Graph Learning (FGL) has emerged as a key paradigm for distributed graph machine learning, enabling cross-domain graph collaborative modeling while preserving data privacy. However, existing methods face two major bottlenecks: the structural heterogeneity discrepancy of graph data among clients weakens the generalization ability of the global model; and model heterogeneity leads to inefficient knowledge sharing and complex global aggregation. To address these issues, we propose FedGVD, an efficient framework that constructs a global perspective through data condensation and server-side virtual node generation, which not only preserves the semantic equivalence of the original data but also avoids privacy leakage. Subsequently, by distributing low-dimensional generalizable knowledge for unidirectional distillation, FedGVD enables local models to absorb global knowledge without transmitting local parameters, thus breaking through the challenges of data and structural heterogeneity as well as model heterogeneity. This innovative approach ensures privacy-preserving and efficient federated graph collaboration. Experiments show that FedGVD maintains excellent performance in heterogeneous model scenarios while significantly improving communication efficiency, offering a new approach for privacy-preserving collaborative modeling in FGL. The code is available at https://github.com/Jasonxx4/FedGVD.

CCS Concepts

• **Computing methodologies** → **Artificial intelligence**; • **Security and privacy**;

Keywords

Federated graph learning; Graph structural heterogeneity; Model heterogeneity; Knowledge distillation

ACM Reference Format:
Zhehao Dai, Guojiang Shen, Yuyue Hu, Jiaxin Du, Xiao Han, and Xiangjie Kong. 2025. FedGVD: Efficient Federated Graph Learning via Unidirectional Distillation with Dynamic Virtual Nodes. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management (CIKM '25), November 10–14, 2025, Seoul, Republic of Korea.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3746252.3761338

*Corresponding author.

CIKM '25, Seoul, Republic of Korea.

ACM ISBN 979-8-4007-2040-6/2025/11
https://doi.org/10.1145/3746252.3761338

1 Introduction

With the rapid advancement of data-driven artificial intelligence technologies, data privacy protection [10, 14] has emerged as a critical issue that demands urgent resolution. Federated learning (FL) [13, 19, 46], as a privacy-preserving distributed training paradigm, enables collaborative modeling among multiple parties without exchanging raw data, thereby achieving knowledge sharing while ensuring privacy. However, in information-sensitive fields such as finance [2] and healthcare [25], data often exists in graph-structured formats, such as account association graphs in financial transaction networks and multimodal disease-drug relationship networks in medical knowledge graphs, where complex interaction patterns are revealed through node attributes and topological connections. The complex graph topologies and node dependencies pose challenges to traditional FL approaches-architectures originally designed for Euclidean data like images and text [13, 23, 29] cannot be directly applied to non-Euclidean graph data. This has given rise to the research field of federated graph learning.

Federated Graph Learning (FGL) [6, 9], as an extension of FL in the realm of graph data, enables collaborative modeling of cross-domain graph data while protecting data privacy. The core objective of FGL is to uncover relational patterns and knowledge embedded in graph data distributed across different institutions or devices through a distributed learning framework [11, 42], thereby supporting tasks such as node classification [12, 22], link prediction [4], and community discovery [1].

However, FGL must not only address data heterogeneity issues inherent in traditional FL [20, 39, 41, 43], such as skewed data distributions and non-IID data, but also confront two major challenges: graph structural heterogeneity [15] and model heterogeneity [32]. Specifically, graph structural heterogeneity manifests in two dimensions: variations in homophily levels across client subgraphs [21] and missing cross-subgraph connections [45]. As illustrated on the top-right in Figure 1, subgraphs from different clients exhibit distinct homophily levels that deviate from the global homophily level, leading to degraded model performance after global training. Additionally, edges that should exist between nodes across clients are lost during data partitioning, compromising the topological integrity of the graph. This dual heterogeneity distorts feature propagation paths. For instance, in regions with missing edges, if only FedAvg [23] is used to aggregate a single type of graph neural network model, the attention mechanism in GATs [27] may erroneously focus on local noisy features, while GCNs [17] with fixed-weight propagation tend to accumulate structural biases. Existing methods like FedSage+ [45] and FGSSL [12] attempt to address missing edges through neighborhood generation but fail to effectively resolve the semantic shift caused by structural heterogeneity.

Model heterogeneity [18, 32, 44] constitutes the second major challenge in FGL. As shown on the lower-right in Figure 1, when different clients adopt heterogeneous GNN architectures (e.g., GCN, GAT, GraphSAGE) due to variations in computational resources or business requirements, fundamental differences in feature propagation paradigms and parameter spaces trigger gradient misalignment. For example, GAT [27] models propagate features through dynamic attention weights, while GraphSAGE [8] employs fixed sampling strategies. Their parameter update directions create conflicts in the vector space during global aggregation. Averaging model parameters—common in traditional FL—leads to performance degradation or catastrophic performance collapse due to mismatches in parameter counts or model scales. Furthermore, conventional parameter-sharing methods incur excessive communication overhead as model sizes grow, rendering them impractical for real-world deployment.

To address the issues of graph structure and model heterogeneity, we propose FedGVD, a communication-efficient and lightweight knowledge distillation framework. In the first round of local communication, we employ data condensation techniques to lighten the knowledge representation while alleviating the problem of different levels of subgraph homogeneity in graph structure heterogeneity. Subsequently, we innovatively design a subgraph integration node on the server side. This virtual node constructs a global-view graph using the locally condensed subgraph data. This mechanism effectively addresses the challenges posed by the lack of cross-client connections in distributed subgraph data by coordinating structural differences among different clients. To tackle the problem of model heterogeneity, we propose a unidirectional distillation framework guided by global knowledge, which can efficiently transfer knowledge from the global model to the local models in a unidirectional manner. The main contributions of this paper are as follows:

- We propose FedGVD, a framework to address graph structural heterogeneity and model heterogeneity in FGL. It achieves cross-domain collaborative modeling of graph data with low communication overhead without requiring model parameter sharing.

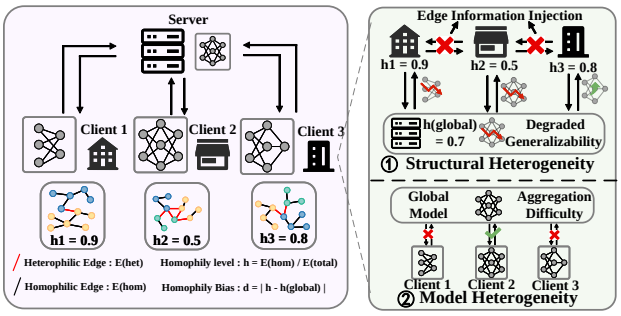


Figure 1: The left side of the figure illustrates the workflow of standard FGL, where each client possesses its own independent subgraph data and model. This setup introduces the two key challenges in FGL shown on the right: graph structural heterogeneity and model heterogeneity.

- We design a virtual node-based mechanism to reconstruct a global topological structure on the server side. This alleviates intra-subgraph homophily level discrepancies and cross-subgraph edge missing issues by dynamically integrating distributed subgraphs.
- We propose a global knowledge-guided unidirectional distillation mechanism that transfers only low-dimensional generalized knowledge, enabling efficient knowledge migration from the global model to heterogeneous local models. This resolves the gradient mismatch problem inherent in model heterogeneity.
- FedGVD demonstrates superior performance across eight widely used graph datasets.

2 Preliminaries

2.1 Problem Definition

FGL is a distributed learning paradigm that enables collaborative training of a global graph model across multiple clients while preserving data privacy. Its core scenario can be formally defined as follows: there exists an implicit global graph $G_g = (\mathcal{V}_g, \mathcal{E}_g)$, where nodes and edges are distributed across different clients. The k-th client holds a subgraph $G_k = (\mathcal{V}_k, \mathcal{E}_k)$, satisfying $v_k \subseteq v_g$ and $\mathcal{E}_k \subseteq \mathcal{E}_g$. Without sharing raw data, clients collaborate with a server to train a GNN model f_{global} while optimizing local models.

A typical approach combines GNNs with the FedAvg framework. In each communication round, the server aggregates model parameters. The global model is then broadcast to participating clients for the next round. The global model aggregation formula is:

$$\widetilde{w}^{t+1} = \sum_{k \in S_t} \frac{|\mathcal{V}_k|}{N} w_k^t, \quad N = \sum_{k \in S_t} |\mathcal{V}_k|, \tag{1}$$

where S_t denotes the set of clients participating in round t, w_k^t represents the parameters of the k-th client's local model, and $|\mathcal{V}_k|$ is the number of nodes in the k-th client's subgraph.

Each client $k \in S_t$ downloads the global model parameters $\widetilde{w}^{t+1}$ and optimizes its local loss function on its subgraph G_k:

$$w_k^{t+1} = \arg\min_w \mathcal{L}(G_k; w). \tag{2}$$

This iterative process aims to balance privacy preservation with collaborative learning across distributed graph data.

2.2 Graph Homophily Level

The homophily level h is defined as the proportion of edges in a graph that connect nodes with the same label. It is defined as:

$$h = \frac{\sum_{(u,v)\in E} I(y_u = y_v)}{|E|}, \tag{3}$$

where E is the set of edges, y_u and y_v are the labels of nodes u and v, and $I(\cdot)$ is an indicator function that returns 1 if $y_u = y_v$ and 0 otherwise. In FGL, clients' local graph data exhibit inconsistent homophily levels (h). According to FedSPA [33], this issue can be divided into two aspects: (1) Homophily Conflict: This arises when the homophily levels of different clients are inconsistent, leading to discrepancies in feature propagation schemes. These discrepancies create conflicts during collaborative training, degrading the generalizability of the global model. (2) Homophily Bias: This is defined as the difference between the homophily level of a client and the global target homophily level. It can be defined as $d_k = |h_k - h_g|$, where h_k is the homophily level of client k, and h_g is the homophily level of the global target. Clients with smaller homophily biases are more aligned with the global optimization direction and thus contribute more significantly to the global model.

2.3 Graph Data Condensation

Data condensation is a technique that generates high-fidelity synthetic datasets to replace raw data. It aims to address issues such as data privacy, storage, and computational efficiency. Its core objective can be formalized as follows: Given a raw dataset $\mathcal{D}_{\text{raw}} = \{(x_i, y_i)\}_{i=1}^{N}$, where x_i are data samples and y_i are their corresponding labels, the goal of data condensation is to generate a condensed dataset $\mathcal{D}_{\text{cond}} = \{(x'_j, y'_j)\}_{j=1}^{M}$(with $M \ll N$). Models trained on D_{cond} achieve performance comparable to or better than those trained on D_{raw}.

Data condensation achieves this by optimizing the generation process to ensure that the synthetic data is equivalent to the raw data for model training purposes. Below are two common techniques used in data condensation:

(1) Gradient Matching: Gradient matching minimizes the distance between the gradient spaces of the synthetic data and the raw data. The objective can be formulated as:

$$\min_{\mathcal{D}_{\text{cond}}} \mathbb{E}_{\theta\sim p(\theta)} \left[\|\nabla_\theta \mathcal{L}_{\text{task}}(\theta, \mathcal{D}_{\text{raw}}) - \nabla_\theta \mathcal{L}_{\text{task}}(\theta, \mathcal{D}_{\text{cond}})\|^2 \right], \tag{4}$$

where $p(\theta)$ is the distribution of model parameters, and L_{task} is the task-specific loss function.

(2) Meta-Learning: Meta-learning employs a bilevel optimization framework, The inner loop optimizes the model parameters θ based on the condensed dataset D_{cond}. The outer loop optimizes the condensed dataset D_{cond} to ensure that the model trained on D_{cond} performs well on the raw dataset D_{raw}. The objective can be formulated as: $\min \mathcal{L}_{\text{task}}(\theta_{\text{cond}}, \mathcal{D}_{\text{raw}})$.

3 Method

Figure 2 provides an overview of the proposed FedGVD framework. First, each client performs parameter-agnostic graph condensation to generate lightweight subgraphs via feature distribution alignment and knowledge distillation (Section 3.1). The server then integrates these heterogeneous subgraphs using a virtual node mechanism, dynamically connecting and harmonizing features to construct a global perspective and mitigate cross-subgraph edge missing issues (Section 3.2). Finally, a globally guided unidirectional distillation framework enables the server to transfer low-dimensional, generalizable knowledge back to clients, supporting efficient knowledge transfer across heterogeneous models and reducing communication and privacy risks (Section 3.3).

3.1 Local Data Condensation

To address the issue of fluctuations in homophily within subgraphs caused by heterogeneous graph structures across clients, we employ the graph condensation method, we use a semantic alignment-based graph condensation method inspired by SimGC [36]. This method compresses raw graph data G_k into a lightweight condensed graph G'_k while retaining key task-related semantic information. The core steps are as follows:

We first randomly initialize a learnable condensed node feature matrix X'. The size of the matrix X' can be selected by the condensation ratio. We define the adjacency matrix A', parameterizing it as a function of X', with the formula given as follows:

$$a'_{ij} = \text{Sigmoid}\left(\frac{1}{2} \cdot \left(MLP([\mathbf{X}'_i; \mathbf{X}'_j]) + MLP([\mathbf{X}'_j; \mathbf{X}'_i])\right)\right),$$

$$\mathbf{A}'_{ij} = \begin{cases} a'_{ij}, \text{if} a'_{ij} \geq \delta \\ 0, \text{otherwise}, \end{cases}$$

where $[\mathbf{X}'_i; \mathbf{X}'_j]$ denotes the concatenation of the i-th and j-th nodes features, δ is a hyperparameter that controls sparsity.

To ensure performance while minimizing computational overhead, we use a pre-trained M-layer Simple Graph Convolutional (SGC) network [35] as a teacher model. Its propagation process is $\mathbf{H}^{(k)} = \hat{\mathbf{A}}'\mathbf{H}^{(m-1)}\mathbf{W}^{(m)}$, where $\hat{A}'$ is the normalized adjacency matrix. By optimizing cross-entropy loss on local data, the teacher model extracts multi-hop neighbor semantic features. After training, the model is frozen, and its layer-wise output features $\mathbf{H}^{(m)}$ serve as semantic alignment targets for the condensed graph.

Semantic equivalence is achieved by jointly optimizing the following objectives: (1) Representation Alignment Loss: Minimizes mean and variance differences of feature distributions across layers.

$$\mathcal{L}_{\text{ra}} = \sum_{m=1}^{M} \lambda_m (\| \mu_m - \mu'_m \|_2^2 + \| \sigma_m - \sigma'_m \|_2^2), \tag{5}$$

where μ_m,σ_m is the mean and standard deviation of the node features of the m-th layer of the original graph, μ'_m,σ'_m is the statistic of the condensed graph, and λ_m is the inter-layer weight coefficient.

(2) Categorization loss: CE loss for labeled condensed nodes.

$$\mathcal{L}_{\text{c}} = \sum_{i\in N'} -Y'_i \log \hat{Y'}_i, \tag{6}$$

where N' denotes the set of nodes in the condensed graph, Y'_i denotes the true label of the i-th node in the condensed graph and $\hat{Y'}$ denotes the predicted probability distribution of the i-th node in the condensed graph.

(3) Feature Smoothness Loss: Ensure that neighboring nodes in the condensed graph have similar characteristics, thus maintaining

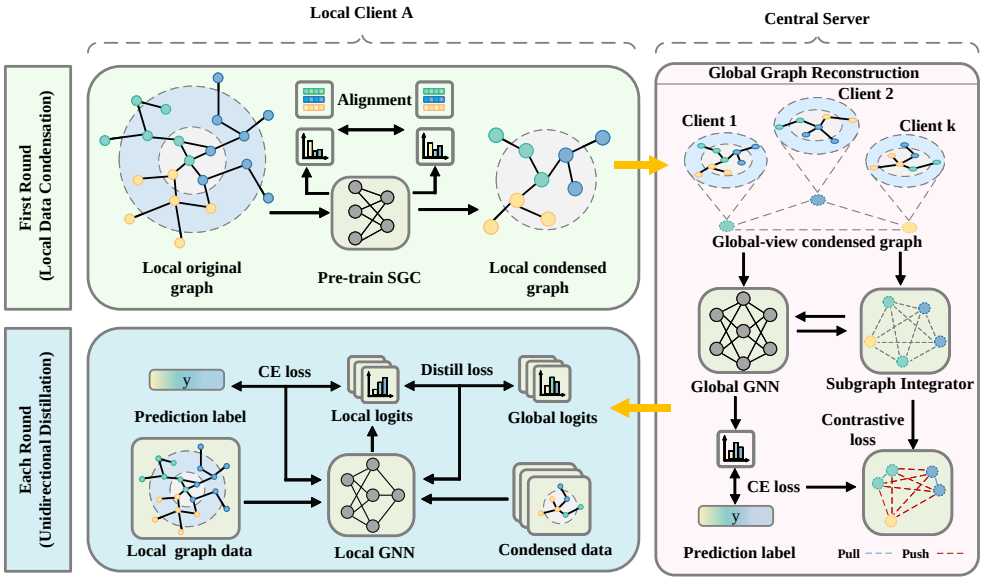


Figure 2: The overview of our FedGVD framework.

the homogeneity property of the graph.

$$\mathcal{L}_{\text{fs}} = \sum_{i,j=1}^{N'} \mathbf{A}'_{ij} \langle \varphi(\mathbf{X}'_i), \varphi(\mathbf{X}'_j) \rangle, \tag{7}$$

where φ denotes the mapping to a higher-dimensional space via the function. The final optimization objective is a weighted sum of these three loss functions, forming a multi-loss optimization target.

$$\mathcal{L}_{\text{total}} = \alpha \mathcal{L}_{\text{ra}} + \beta \mathcal{L}_{\text{c}} + \gamma \mathcal{L}_{\text{fs}}. \tag{8}$$

By alternately optimizing the parameters of the condensed graph and the teacher model, efficient data compression is achieved while preserving the key semantic information of the original graph.

3.2 Global graph reconstruction

To address the issue of missing edges across subgraphs, we perform global graph reconstruction on the server side. We introduce the Subgraph integrator, a virtual node that establishes connections and enables alignment between different subgraphs. While preserving the unique structure of each subgraph, it facilitates information flow and knowledge sharing across subgraphs. This allows the global model on the server side to capture a holistic perspective and train using condensed subgraph data.

Server-side assigns a subgraph integrator V_k to each client's condensed subgraph G'. This node establishes connections with all nodes within the subgraph, and its initial features are aggregated from the features of the corresponding subgraph. The formula is as follows: $\mathbf{x}_k = \text{MLP}\left(\sum_{v \in V_k} \alpha_{vk} \mathbf{x}_v\right)$, where V_k is the set of nodes of the k-th client, α_{vk} is the importance weight of node v for subgraph integrator k, calculated by cosine similarity:

$$\alpha_{vk} = \frac{\exp(\sin(\mathbf{q}, \mathbf{x}_v))}{\sum_{u \in V_k} \exp(\sin(\mathbf{q}, \mathbf{x}_u))}, \tag{9}$$

where $\mathbf{q}$ is the learnable query vector. This design ensures that the subgraph integrator can represent most of the features within the subgraph and enables efficient communication and information exchange with its corresponding subgraph.

Furthermore, we dynamically connect the subgraph integrators, allowing isolated subgraphs to exchange information. This not only connects more nodes with high similarity but also alleviates the problem of homogeneity and heterogeneity. Additionally, it enables the server to reconstruct a graph, providing a global perspective.

Specifically, the subgraph integrators are connected using coefficient similarity, dynamically establishing global associations. The cosine similarity matrix between all virtual nodes is computed:

$$S_{k_1,k_2} = \frac{x_{k_1} \cdot x_{k_2}}{\| x_{k_1} \| \| x_{k_2} \|}, S_{k_1,k_2} \in \mathbb{R}^{K \times K}. \tag{10}$$

Here, x_{k_1},x_{k_1} denotes the feature vector of the k_1 and the k_2 subgraph integrator, and K is the number of clients. For each client k, only the T most relevant connections are retained. This ensures that each client is connected to only a few of the most relevant clients, accelerating global model training and inference. The feature matrix of the global graph is formulated as follows:

$$\tilde{\mathrm{X}}' = \begin{bmatrix} \mathrm{X}'_{diag} & \mathrm{X}_{cross} \\ \mathrm{X}_{cross} & \mathrm{X}_{integer} \end{bmatrix}, \tag{11}$$

where X'_{diag} is a diagonal matrix containing the adjacency matrices of individual graph datasets, X_{cross} is the matrix for cross-graph connections, and $\mathrm{X}_{integer}$ is the matrix for connections between integrators. The subgraph integrators dynamically update with the global model, so the similarity matrix between integrators also updates dynamically. This ensures that the global graph in FL continuously adapts to the evolution of client data.

The global model is trained with a multi-task loss function, including contrast loss, reconstruction loss, and classification loss:

$$\mathcal{L}_{\text{global}} = \mathcal{L}_{\text{contrastive}} + \mathcal{L}_{\text{classification}} \tag{12}$$

Contrastive loss $\mathcal{L}_{contrastive}$ is used to maximize the feature similarity of the relevant integrator nodes:

$$\mathcal{L}_{\text{contrastive}} = -\log \frac{\exp(\sin(\mathbf{x}_{k_1}, \mathbf{x}_{\text{global}})/\tau)}{\sum_{k_1 \neq k_2} \exp(\sin(\mathbf{x}_{k_1}, \mathbf{x}_{k_2})/\tau)}, \tag{13}$$

where $\mathbf{x}_{\text{global}}$ is the server-generated global feature representation, for capturing patterns and features common to all client subgraph integrator nodes, $\sin(\mathbf{x}_{k_1}, \mathbf{x}_{k_2})$ denotes the pair of subgraph integrator nodes from different clients, and τ denotes the temperature parameter. The classification loss aims to drive the global model to correctly predict the classes of the subgraph integrator nodes. Formally, it is defined as:

$$\mathcal{L}_{\text{classification}} = \text{CE}(\mathbf{y}, \text{softmax}(W\mathbf{x}_k)), \tag{14}$$

where W denotes the weight matrix for mapping the features of the subgraph integrator nodes to the category space. The features of the subgraph integrator are dynamically updated through end-to-end gradient descent: $\mathbf{x}_k \leftarrow \mathbf{x}_k - \eta \nabla_{\mathbf{x}_k} \mathcal{L}_{\text{global}}$, η denotes learning rate. Overall, global graph reconstruction achieves cross-client knowledge sharing and efficient collaboration through dynamic updates of subgraph integrators and multi-loss optimization. This approach not only effectively addresses the issues of data isolation and missing edges across subgraphs among clients but also significantly enhances the performance and communication efficiency of FGL.

3.3 Unidirectional distillation framework

To address the challenges of model heterogeneity-where clients employ diverse GNN architectures, resulting in issues such as gradient misalignment, parameter incompatibility, increased communication overhead, and heightened privacy risks-we propose a unidirectional distillation framework guided by global knowledge. Instead of sharing model parameters, the server distills low-dimensional, generalizable knowledge from the integrated global model and transmits it to clients. This enables efficient knowledge transfer across heterogeneous local models, reducing communication overhead and preserving privacy.

In the first round of federated information exchange, we send condensed data from clients and low-dimensional universal knowledge logits computed by the global model as a data pair to each client. In subsequent communication processes, only the corresponding logits need to be transmitted to guide the learning of heterogeneous client models for global topological semantics, achieving communication-efficient knowledge distillation. Client k receives the content as follows:

$$\mathcal{T}_k = \{z_n \mid z_n = f_{\text{global}}(\tilde{X}_n)\}_{n=1}^{K} \cup \{\tilde{X}_n\}_{n \neq k},$$

where z_n denotes logits generated by the global model f_{global} for client k's condensed data $\tilde{X}_n$. Afterward, clients perform local training optimization, leveraging global logits and cross-client condensed to data enhance model performance. We define the local classification loss as:

$$\mathcal{L}_{\text{task}} = \text{CE}\left(f_{\text{local}}(X_k), y_k\right) + \text{CE}\left(f_{\text{local}}(\tilde{X}_k), y_k^{\text{syn}}\right). \tag{15}$$

Subsequently, we perform spatial normalization on the local logits and the global logits to eliminate scale differences in the outputs of heterogeneous models, ensuring fair knowledge transfer.We soften the probability distribution to enhance the model's ability to learn inter-class associations. The formulas are as follows:

$$z_k^{\text{global}} = \frac{z_k - \mu_{\text{global}}}{\sigma_{\text{global}}}, \quad z_k^{\text{local}} = \frac{f_{\text{local}}(\tilde{X}_k) - \mu_{\text{local}}}{\sigma_{\text{local}}},$$

where μ denotes the mean of global or local Logits, σ denotes the standard deviation of global or local Logits. In this way all Logits are mapped to the space of zero mean and unit variance. We define the local knowledge distillation loss function as:

$$\mathcal{L}_{\text{distill}} = \sum_k \text{KL}\left(\text{softmax}(z_k^{\text{global}}/\tau) \parallel \text{softmax}(z_k^{\text{local}}/\tau)\right), \tag{16}$$

where τ denotes the temperature coefficient. The KL divergence term encourages the local model to learn class correlations from the global view. The overall local loss function is defined as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \omega \mathcal{L}_{\text{distill}}, \tag{17}$$

where ω is a balancing parameter that controls the contributions of the distill loss function. The entire process only requires knowledge transmission from the server to clients, eliminating the need for clients to feedback parameters or prototypes. By transmitting low-dimensional universal knowledge, the impact of model heterogeneity is mitigated, while ensuring a lightweight design and privacy protection for the framework. The pseudo-code for the training process of our method is in Algorithm 1.

4 Experiments

4.1 Experimental settings

Datasets. In this paper, we conducted experiments on eight commonly used graph datasets: five datasets with high homophily (Cora, CiteSeer, PubMed, Amazon-Photo, and Amazon-Computers) [30, 40] and three datasets with low homophily (Chameleon, Actor, and Amazon-Ratings) [26, 28]. Further details are provided in Table 1. To simulate the client subgraph distributions in FGL, we employed the Louvain algorithm to achieve subgraph division for each client. This algorithm is widely adopted for subgraph partitioning in FGL due to its efficient community detection capabilities.

Baselines. We compared FedGVD with several state-of-the-art approaches: two conventional FL methods: FedAvg [23], FedDC [7]; two prototype-based learning methods: FedProto [32], FedTGP [44]; seven graph structure-aware FGL methods: FedSage+ [45], FedPUB [1], FedGTA [22], FGSSL [12], FedGL [3], AdaFGL [21] and FedTAD [47]; one personalized method: FGGP [34].

Implement details. The experiments were conducted within the OpenFGL [38] framework. In the main study under model homogeneous conditions, we adopted 2-layers GCN [17] as the backbone network for both clients and the central server. The hidden layer dimension was set to 128. The number of local training epochs and total communication rounds were configured as 3 and 100, respectively. The Adam optimizer [16] was employed with a local model learning rate of 0.01 (adjusted to 0.001 in some experiments), weight decay of $5e^{-4}$, and dropout rate of 0.5. The data condensation ratio is set to 0.2 by default, and the hyperparameter for data distillation is set to $1e^{-4}$. Under model-heterogeneous conditions (HeFE5) with 10 clients, we utilized five distinct 2-layers model architectures: GCN [17], GAT [27], SGC [35], MLP, and GraphSAGE [8], while maintaining the same hidden layer dimension. The evaluation

Algorithm 1: FedGVD algorithm

Input: K clients with local subgraphs G_k, Communication rounds T, Temperature τ
Output: Global model f_{global}, Client models f_k

ServerExecutes:
Initialize virtual node Subgraph Integrator set $V = \emptyset$
// Phase 1: Global graph construction
for $t = 1$ *to* T **do**
  if $t == 1$ **then**
    for *each client* $k \in [K]$ **do**
      $G'_k, \theta_k \leftarrow \text{Condensation}(G_k, \gamma)$ // Phase 3
    end
    Server aggregates G'_k and constructs
    Subgraph integrator $V = v_k$ via Eq.(9)
    Global graph G_{global} via Eq.(11)
  end
  Train f_{global} on G_{global} and V via Eq.(12-14)
  // Phase 2: Knowledge distillation
  Generate global logits $\{z_k\} = f_{global}(G_{global})$
  for *each client* $k \in [K]$ **do**
    ClientUpdate(k, z_j, G'_j) // Phase 4
  end
end
ClientLocalTraining:
for $t = 1$ *to* T **do**
  if $t == 1$ **then**
    // Phase 3: Local data condensation
    Pre-train SGC on G_k
    while *not converged* **do**
      Calculate loss via Eq.(5-8)
      Update X', A' via gradient descent
    end
    return $G'_k = (X', A')$, θ_k
    Upload G'_k to Server
    Receive $\tilde{X}_n$ from Server
  end
  // Phase 4: Unidirectional distillation
  Receive z_n from Server
  $z_{global} = \text{normalize}(z_n)$
  $z_{local} = \text{normalize}(f_k(\tilde{X}_n))$
  Calculate loss via Eq.(15-17)
end

Table 1: Dataset statistics

Dataset	#Nodes	#Edges	#Classes	#E.Homo
Cora	2,708	5,429	7	0.81
CiteSeer	3,327	4,732	6	0.74
PubMed	19,717	44,338	3	0.80
Amazon-Photo	7,487	119,043	8	0.83
Amazon-Computers	13,381	245,778	10	0.78
Chameleon	2,277	36,101	5	0.23
Actor	7,600	29,926	5	0.22
Amazon-ratings	24,492	93,050	5	0.38

metric was accuracy of the node classification task. To evaluate the robustness of our results under different initial conditions, we conducted three independent replicate trials for all experimental configurations. By calculating the mean and standard deviation of the prediction performance metrics, we effectively eliminated the random bias in the model training process, ensuring the statistical significance of the evaluation results.

Experiment Environment. The experimental machine with Intel(R) Xeon(R) Gold 6226R CPU @ 2.90GHz, and NVIDIA GeForce RTX 3090 with 24GB memory and CUDA 12.4.

4.2 Performance analysis

4.2.1 Performance comparison. We present the test accuracy of all methods across eight datasets in Table 2. Specifically, we evaluate the model accuracy of each algorithm under a 10-client setup. The results demonstrate that FedGVD achieves superior performance in the HtFE5 heterogeneous scenario, consistently outperforming all baseline methods with an average test accuracy that exceeds the baseline by 3.26%. Additionally, in the homogeneous model scenario, FedGVD achieves the best performance, significantly surpassing the baseline methods with an average test accuracy improvement of 3.69% over the baseline. Notably, FedGVD shows even more pronounced performance improvements in datasets with stronger data heterogeneity, with an average enhancement of 4.64%. This result highlights the effectiveness of our proposed subgraph integrator and unidirectional knowledge distillation approach in addressing both homogeneity and heterogeneity challenges.

4.2.2 Impact of client numbers. To verify the scalability of FedGVD in settings with more clients, we conducted experiments on the Cora, Amazon-Computers, and Actor datasets using Louvain partitioning, under both model homogeneous and HtFE5 heterogeneous scenarios with 10, 20, and 30 clients. As shown in Table 3, model accuracy decreases as the number of clients increases due to less local data and increased data heterogeneity. However, FedGVD maintains significant superiority, outperforming baselines by an average of 4.30%. Notably, even in heterogeneous scenarios, FedGVD surpasses baseline methods under homogeneous conditions.

4.2.3 Impact of model heterogeneity. We further evaluate the performance of FedGVD across five scenarios with increasing levels of model heterogeneity, including homogeneous, HtFE2, HtFE4, and HtFE5 scenarios. Specifically, HtFE2 includes GCN and GAT; HtFE4 includes GCN, GAT, SGC, and GraphSAGE; HtFE5 includes GCN, GAT, SGC, GraphSAGE, and MLP. Experiments were conducted on the Cora, Computers, and Chameleon datasets with 20 clients, and the results are detailed in Table 4. Specifically, as model heterogeneity increases, the performance of both FedTGP [44] and FedGVD decreases. FedTGP's accuracy drops by at least 4.11% from the homogeneous to the HtFE5 scenario. In contrast, FedGVD only declines by 0.75% and remains 8.30% ahead of the baseline.

4.2.4 Communication cost. To verify that our proposed method, FedGVD, has lower communication overhead, We conducted experiments on the Chameleon dataset under the homogeneous model scenario to calculate the total communication consumption per round for ten clients, which is divided into upload and download data, as shown in Table 5. Specifically, although the communication

Table 2: Performance comparison between FedGVD and baseline models across eight datasets with 10 clients each. The best results are highlighted in bold, while the second-best results are underlined, and ± indicating standard deviations.

Method	Cora	CiteSeer	PubMed	Photo	Computers	Chameleon	Actor	Ratings
GCN-Local	78.31±0.25	67.25±0.64	84.48±0.46	90.59±0.48	87.39±0.34	59.47±0.63	27.28±1.24	46.71±0.92
GAT-Local	78.92±0.63	64.63±0.82	82.77±0.63	90.36±0.57	87.19±0.36	58.57±0.52	27.75±0.83	46.35±0.65
FedAvg	82.41±0.61	69.59±0.56	85.49±0.32	90.68±0.82	87.56±0.51	56.78±1.08	28.38±0.69	44.87±0.33
FedDC	82.41±0.47	70.57±0.47	83.08±0.56	89.18±0.35	77.74±0.21	54.68±0.92	30.77±1.32	37.80±0.47
FedProto	81.12±0.54	64.77±0.33	84.29±0.68	87.97±0.53	84.14±0.49	57.99±0.62	28.38±0.64	39.56±0.31
FedTGP	81.31±0.69	65.73±0.42	84.43±0.21	87.34±0.45	78.38±0.45	60.52±0.57	27.49±0.75	46.31±0.54
FedSage+	83.34±0.74	70.11±0.97	86.30±0.76	91.68±1.18	84.29±0.83	59.95±0.85	30.48±1.14	46.62±0.67
FedGTA	82.70±0.43	69.65±0.62	85.38±0.48	91.29±0.31	86.54±0.65	50.37±0.40	29.11±0.58	44.78±0.27
Fed-PUB	83.77±0.62	71.16±0.83	86.47±0.53	90.33±0.67	88.98±0.27	60.08±0.64	29.42±0.97	43.26±0.83
FGSSL	83.62±0.58	69.47±0.71	85.51±0.32	88.22±0.73	88.21±0.63	59.83±0.82	28.01±0.83	44.74±0.66
FedGL	78.43±0.75	70.82±0.96	82.42±0.79	89.24±0.47	88.73±0.76	60.11±0.93	27.23±1.23	45.10±0.61
AdaFGL	84.53±0.81	69.58±0.69	84.44±0.72	91.93±0.92	85.89±0.68	57.07±1.14	30.78±1.39	45.81±0.58
FGGP	80.49±0.39	69.18±0.51	86.46±0.49	87.38±0.65	87.62±0.36	57.67±0.76	28.49±0.82	39.45±0.52
FedTAD	84.26±0.61	69.81±0.96	85.50±0.61	91.21±0.51	86.17±0.52	56.26±1.22	28.43±1.05	44.91±0.90
FedGVD (ours)	**85.56±0.29**	**71.18±0.61**	**86.71±0.37**	93.15±0.43	**90.18±0.39**	62.07±0.60	32.04±0.83	**49.98±0.54**
FedGVD (HtFE5)	84.47±0.35	69.51±0.56	86.28±0.42	**93.43±0.47**	89.36±0.34	**62.35±0.61**	**32.43±0.77**	49.67±0.59

Table 3: Performance comparison with methods under varying client numbers. The best and second-best results are highlighted in bold and underlined, respectively, with ± indicating standard deviations.

Method	Cora			Computers			Chameleon		
	10	20	30	10	20	30	10	20	30
GCN-Local	78.31±0.25	76.89±0.30	75.24±0.36	87.39±0.34	86.44±0.40	84.81±0.42	59.47±0.63	52.26±0.58	43.13±0.51
GAT-Local	78.92±0.63	75.72±0.54	74.22±0.58	87.19±0.36	84.14±0.33	83.47±0.28	58.57±0.52	49.00±0.61	44.59±0.59
FedAvg	82.41±0.61	77.58±0.73	75.92±0.69	87.56±0.51	85.77±0.46	85.40±0.35	56.78±1.08	50.07±1.18	45.89±0.92
FedDC	82.41±0.47	81.53±0.41	77.84±0.52	77.74±0.21	84.51±0.27	83.65±0.39	54.68±0.92	52.25±0.88	47.19±0.81
FedProto	81.12±0.54	78.24±0.70	74.69±0.66	84.14±0.49	83.53±0.53	81.24±0.50	57.99±0.62	50.92±0.69	43.79±0.73
FedTGP	81.31±0.69	78.52±0.59	74.23±0.56	78.38±0.45	79.27±0.37	77.97±0.53	60.52±0.57	50.09±0.63	45.11±0.69
FedSage+	83.34±0.74	77.22±0.80	74.78±0.62	84.29±0.83	84.62±0.88	86.64±0.93	59.95±0.85	50.47±0.89	45.73±0.73
FedGTA	82.70±0.43	79.25±0.41	75.91±0.47	86.54±0.65	85.81±0.60	85.73±0.73	50.37±0.40	50.62±0.49	45.12±0.52
Fed-PUB	83.77±0.62	79.81±0.66	75.65±0.71	88.98±0.27	87.01±0.33	86.37±0.38	60.08±0.64	51.74±0.56	46.50±0.52
FGSSL	83.62±0.58	79.32±0.51	74.54±0.66	88.21±0.63	86.33±0.69	84.73±0.76	59.83±0.82	50.16±0.80	45.37±0.71
FedGL	78.43±0.75	79.63±0.79	75.83±0.83	88.73±0.76	86.50±0.69	85.21±0.80	60.11±0.93	51.83±0.89	46.42±0.72
AdaFGL	84.53±0.81	79.07±0.88	75.80±0.84	85.89±0.68	86.82±0.77	85.08±0.72	57.07±1.14	52.25±0.99	48.46±1.06
FGGP	80.49±0.39	78.97±0.44	75.58±0.36	87.62±0.36	85.18±0.33	84.88±0.29	57.67±0.76	50.49±0.92	47.28±0.87
FedTAD	84.26±0.61	79.17±0.79	76.21±0.64	86.17±0.52	85.29±0.60	85.99±0.55	56.26±1.22	51.33±1.16	47.07±1.08
FedGVD (ours)	**85.56±0.29**	**83.54±0.33**	**80.24±0.35**	**90.18±0.39**	**89.03±0.43**	**87.74±0.48**	62.07±0.60	**55.30±0.52**	**51.27±0.58**
FedGVD (HtFE5)	84.47±0.35	82.63±0.30	79.33±0.39	89.36±0.34	88.33±0.31	87.21±0.40	**62.35±0.61**	54.66±0.55	51.18±0.50

Table 4: Performance under different degrees of model heterogeneity

Dataset	Method	Different Model Heterogeneity			
		Isomorphic	HtFE2	HtFE4	HtFE5
Cora	FedProto	78.52	77.03	76.34	76.23
	FedTGP	78.52	77.03	76.34	76.23
	FedGVD	**83.54**	**83.32**	**83.48**	**82.63**
Chameleon	FedProto	50.92	50.37	48.83	47.54
	FedTGP	50.09	48.32	47.89	47.26
	FedGVD	**55.30**	**55.04**	**54.96**	**54.66**

consumption of FedGVD is relatively high in the first round, it is still significantly lower than methods that use model parameter sharing, such as FedAvg and FedSage+. In subsequent rounds, FedGVD does not require uploading data, and its download consumption is the lowest among all methods. After averaging, our method achieves a communication consumption of only 0.01MB per round, which is the lowest among all methods, while still maintaining the best performance. This highlights the effectiveness of our approach in knowledge transfer for FGL.

4.2.5 Impact of local training rounds. We experimented on the Computers dataset with 20 clients, setting E to 3, 5, 7, and 10. As shown in Table 6, most methods saw improved accuracy with increasing E. Notably, FedGVD maintained excellent performance across different E values. In the case of model homogeneity, FedGVD achieved high accuracy even at E=3, indicating its ability to effectively leverage local updates to enhance model performance while reducing communication rounds.

4.2.6 Convergence performance. As visualized in the convergence curves of Figure 3 on four datasets, FedGVD outperforms other baseline methods by maintaining higher initial performance and achieving faster and more stable convergence in both homogeneous

Table 5: Communication Cost (MB) and Accuracy (%) on Chameleon

Method	Upload	Download	Accuracy
FedAvg	11.43	11.43	56.78
FedDC	11.86	11.86	54.68
FedTGP	0.32	0.32	60.52
FedSage+	20.82	11.43	59.95
FedGTA	11.68	11.52	50.37
Fed-PUB	11.85	11.60	60.08
FedGL	12.06	11.82	60.11
FGGP	11.86	11.43	57.67
FedTAD	23.48	11.43	56.26
FedGVD (first round)	8.78	1.84	62.07
FedGVD (each round)	0	0.02	62.35

Table 6: Performance comparison under different local update epochs.

Method	local update epochs			
	E = 3	E = 5	E = 7	E = 10
FedAvg	85.77	86.13	86.34	86.51
FedDC	84.51	84.37	84.55	85.11
FedProto	83.53	84.16	83.94	84.36
FedTGP	79.27	80.23	80.67	80.34
FedSage+	84.62	85.22	85.91	86.27
FedGTA	85.81	86.37	86.98	87.78
Fed-PUB	87.01	87.67	87.88	88.11
FGSSL	86.33	86.84	87.45	87.85
FedGL	86.50	86.67	86.94	87.39
AdaFGL	86.82	85.96	86.78	86.72
FGGP	85.18	85.94	86.27	86.35
FedTAD	85.29	86.20	86.47	85.72
FedGVD	89.03	88.75	88.92	89.04
FedGVD (HtFE5)	88.33	88.35	88.47	88.63

and heterogeneous model settings. This demonstrates that FedGVD is suited for FGL scenarios with limited communication overhead.

4.3 Ablation study.

To systematically evaluate the specific impact of each module on the performance of FedGVD, we progressively removed or replaced key components within the framework. The experiments were conducted under the same default settings as previous experiments to ensure comparability and consistency of the results. Through this approach, we were able to clearly reveal the contributions of these modules to the overall performance of FedGVD. The ablation results are shown in Table 7, and experiments were on the Computers and Cora datasets: (1) Subgraph Integrator: Removing the subgraph integrator prevents the global model from effectively utilizing cross-client topological information, leading to reduced knowledge transfer efficiency. (2) Condensed Data from Other Clients: Without the condensed data from other clients, the local model can only rely on its own data for training and cannot leverage generalized information from the global model, resulting in reduced generalization capability. (3) Unidirectional Knowledge Distillation: Eliminating unidirectional knowledge distillation prevents the local model from effectively absorbing global patterns across clients, leading to reduced generalization capability.

Table 7: Ablation study of key components of FedGVD on Cora and Computers.

SI	CDOC	UKD	Cora			Computers		
			10	20	30	10	20	30
✗	✗	✗	79.25	75.98	71.71	85.56	85.82	84.05
✗	✗	✓	80.13	76.42	72.68	86.87	86.06	84.31
✗	✓	✓	83.70	80.91	77.79	88.06	86.36	85.62
✓	✗	✓	81.39	77.83	74.28	87.78	86.17	84.59
✓	✓	✗	80.24	78.64	74.57	87.67	86.34	85.29
FedGVD			**84.47**	**82.63**	**79.33**	**89.36**	**88.33**	**87.21**

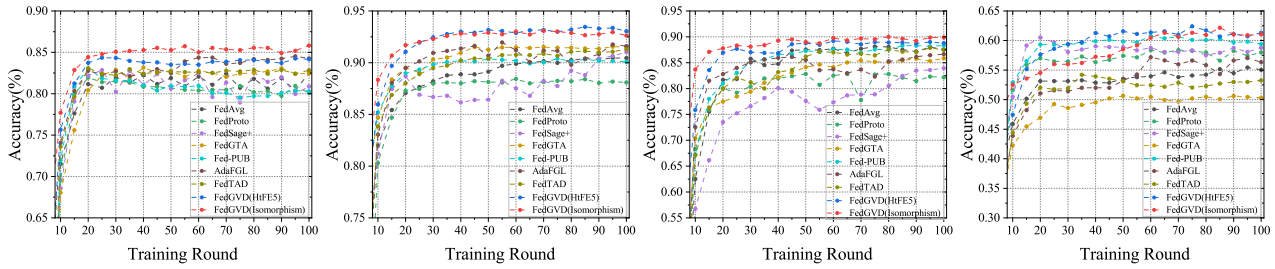
4.4 Sensitivity analysis

4.4.1 Graph condensation rate. To further investigate the impact of the compression ratio on the performance of FedGVD, we conducted experiments to evaluate how varying compression ratios affect model performance. The experiments were carried out in a setup with 10 clients using the rating dataset, and the results are presented in Figure 4(a). The results demonstrate that a higher condensation rate generally correlates with improved model accuracy. However, we observed that increasing the compression ratio from 0.1 to 0.2 leads to a significant performance gain, while further increases in the compression ratio yield only marginal improvements. It is important to note that communication costs increase linearly with the compression ratio. Additionally, a lower compression ratio reduces both transmission costs and enhances local data privacy. Considering the trade-offs between communication costs, performance, and privacy protection, we conclude that a compression ratio of 0.2 strikes the optimal balance among these factors and is therefore the most recommended choice.

4.4.2 Impact of loss function coefficient. The distillation loss function optimizes the student model by imitating the teacher model's output, with its hyperparameter determining the weight of the distillation loss in the total loss. We conducted a sensitivity analysis of the coefficient w of the distillation loss function in distributed learning scenarios with ten clients on the Computers and Photo datasets. The experimental results in Figure 4(b) show that when the hyperparameter value is between 10^{-2} and 10^{-4}, the model maintains good performance, and the optimal performance is achieved when the hyperparameter value is 10^{-4}. Further observations reveal that changes in the hyperparameter have a more pronounced impact on the heterogeneous scenario (HtFE5). This indicates that the guidance of the global model is critical for performance improvement in model-heterogeneous distributed learning scenarios. This phenomenon validates the effectiveness of the unidirectional distillation module in heterogeneous scenarios and provides strong support for knowledge transfer in distributed learning.

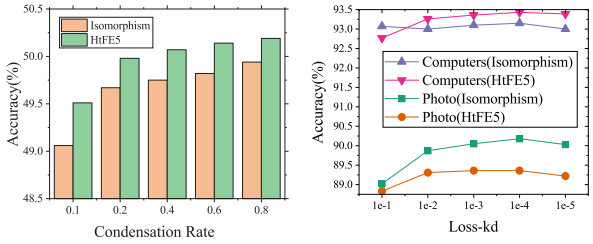
5 Related Work

Federated graph learning. In recent years, FL has gained increasing attention in graph machine learning due to its unique advantages in distributed data privacy protection and efficient collaboration. Current research in this field is primarily divided into two directions: graph-level federated learning and subgraph-level federated learning. In graph-level federated learning, each client holds multiple independent graphs, and downstream tasks typically focus on graph-level classification. For example, GCFL [37]



(a) Cora (b) Photo (c) Computers (d) Chameleon

Figure 3: Comparison of convergence performance between our proposed FedGVD and baseline methods across four graph datasets with 10 participating clients.



(a) Rating (b) Computers and Photo

Figure 4: Impact of graph structure condensation rate and the weight coefficient ω under 10 clients FGL scenario.

introduces a dynamic clustering method based on GNN gradients, while FedStar [31] leverages structural knowledge sharing by combining structural embeddings with feature-structure decoupling, enabling clients to learn domain-agnostic structural knowledge while preserving domain-specific feature knowledge. Subgraph-level federated learning assumes that multiple clients collectively hold subgraphs of a global graph, primarily addressing node or edge-level classification and prediction tasks. FedGTA [22] improves model performance and scalability through local confidence smoothing and personalized aggregation of mixed neighborhood features. FedTAD [47] proposes a topology-aware data-agnostic knowledge distillation method that transfers knowledge reliably through pseudo-graph generation.

Structural heterogeneity in FGL. Under the data heterogeneity challenges of traditional FL, FGL faces additional critical challenges: graph structural heterogeneity, including homophily differences in node connection patterns within subgraphs and missing cross-subgraph edges. FedSPA [33] identifies dynamic heterogeneous characteristics in subgraph data homophily, termed homophily heterogeneity. FGSSL [12] calibrates local model bias via graph-level structural distillation. FED-PUB [1] evaluates subgraph similarity using functional embeddings for weighted aggregation and selectively updates parameters with weight masks. To address missing cross-subgraph edges, FedSage+ [45] introduces a missing neighbor generator to complete missing cross-subgraph links, while FedGL [3] supplements local models by generating a global pseudo-graph from node embeddings to augment training labels and graph structures. Currently, there is a lack of FGL methods that can effectively address both major types of structural heterogeneity.

Model heterogeneity in FGL Model heterogeneity is another challenge in FGL, arising from hardware variations, computational constraints, or personalized requirements. Clients may adopt heterogeneous neural architectures (e.g., GCN, GAT), making direct parameter aggregation infeasible. Current solutions focus on three mainstream approaches: knowledge distillation, prototype learning, and meta-learning. FedDistill [48] utilize knowledge distillation to align local models with the global model without sharing parameters. FedProto [32] and FedProc [24] share low-dimensional generic prototype knowledge to regularize local model training using prototype data. Building on this, FedTGP [44] and FGGP [34] further optimize prototype generation to mitigate prototype domain shift across clients. Per-FedAvg [5] employs a model-agnostic meta-learning (MAML) framework to identify a globally shared initialization. It trains personalized local models derived from the global initialization. Existing knowledge distillation and meta-learning methods incur high communication overhead from frequent parameter transmission, while prototype learning sacrifices structural granularity for class-level statistics.

6 Conclusion

This paper introduces FedGVD, a lightweight FGL algorithm that addresses graph structural heterogeneity and model heterogeneity through a dynamic subgraph integrator and a unidirectional distillation framework. The method reconstructs a graph with global structural awareness on the server side using locally condensed data and performs low-communication knowledge distillation by unidirectionally transmitting condensed data-logits pairs from the server to clients. Experiments demonstrate that FedGVD outperforms state-of-the-art FGL methods in terms of model accuracy, convergence speed, and communication overhead.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 62476247, 62072409 and 62402442, in part by the "Pioneer" and "Leading Goose" R&D Program of Zhejiang under Grant 2024C01214, and in part by the Zhejiang Provincial Natural Science Foundation under Grant LR21F020003 and LQ24F020038.

GenAI Usage Disclosure

During the preparation of this manuscript, ChatGPT was used exclusively for light editing purposes, including grammar correction, sentence rephrasing, and general language polishing. No content was generated by ChatGPT beyond these editorial functions. All ideas, technical content, and writing were originally conceived and authored by the listed authors.

References

[1] Jinheon Baek, Wonyong Jeong, Jiongdao Jin, Jaehong Yoon, and Sung Ju Hwang. 2023. Personalized subgraph federated learning. In *Proc. of ICML*. PMLR, 1396–1415.

[2] David Byrd and Antigoni Polychroniadou. 2020. Differentially private secure multi-party computation for federated learning in financial applications. In *Proceedings of the first ACM International Conference on AI in Finance*. 1–9.

[3] Chuan Chen, Ziyue Xu, Weibo Hu, Zibin Zheng, and Jie Zhang. 2024. FedGL: Federated graph learning framework with global self-supervision. *Information Sciences* (2024), 119976.

[4] Mingyang Chen, Wen Zhang, Zonggang Yuan, Yantao Jia, and Huajun Chen. 2021. Fede: Embedding knowledge graphs in federated setting. In *Proceedings of the 10th International Joint Conference on Knowledge Graphs*. 80–88.

[5] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. 2020. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. *Proc. of NeurIPS* (2020), 3557–3568.

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

[7] Liang Gao, Huazhu Fu, Li Li, Yingwen Chen, Ming Xu, and Cheng-Zhong Xu. 2022. Feddc: Federated learning with non-iid data via local drift decoupling and correction. In *Proc. of CVPR*. 10112–10121.

[8] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Proc. of NeurIPS* (2017).

[9] Chaoyang He, Keshav Balasubramanian, Emir Ceyani, Carl Yang, Han Xie, Lichao Sun, Lifang He, Liangwei Yang, Philip S Yu, Yu Rong, et al. 2021. FedGraphNN: A Federated Learning System and Benchmark for Graph Neural Networks. In *Proc. of ICLR*.

[10] Ming Hu, Zhihao Yue, Xiaofei Xie, Cheng Chen, Yihao Huang, Xian Wei, Xiang Lian, Yang Liu, and Mingsong Chen. 2024. Is aggregation the only choice? federated learning via layer-wise model recombination. In *Proc. of KDD*. 1096–1107.

[11] Ming Hu, Peiheng Zhou, Zhihao Yue, Zhiwei Ling, Yihao Huang, Anran Li, Yang Liu, Xiang Lian, and Mingsong Chen. 2024. FedCross: Towards accurate federated learning via multi-model cross-aggregation. In *2024 IEEE 40th International Conference on Data Engineering*. IEEE, 2137–2150.

[12] Wenke Huang, Guancheng Wan, Mang Ye, and Bo Du. 2023. Federated graph semantic and structural learning. In *Proc. of IJCAI*. 3830–3838.

[13] Wenke Huang, Mang Ye, and Bo Du. 2022. Learn from others and be yourself in heterogeneous federated learning. In *Proc. of CVPR*. 10143–10153.

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

[15] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. 2021. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning* (2021), 1–210.

[16] DP Kingma. 2014. Adam: a method for stochastic optimization. In *Int Conf Learn Represent*.

[17] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *Proc. of ICLR*.

[18] Xiangjie Kong, Haopeng Yuan, Guojiang Shen, Hanlin Zhou, Weiyao Liu, and Yao Yang. 2024. Mitigating data imbalance and generating better prototypes in heterogeneous federated graph learning. *Knowledge-Based Systems* (2024), 111876.

[19] Long Tan Le, Tuan Dung Nguyen, Tung-Anh Nguyen, Choong Seon Hong, Suranga Seneviratne, Wei Bao, and Nguyen H Tran. 2024. Federated Deep Equilibrium Learning: Harnessing Compact Global Representations to Enhance Personalization. In *Proc. of CIKM*. 1132–1142.

[20] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. 2021. Ditto: Fair and robust federated learning through personalization. In *Proc. of ICML*. PMLR, 6357–6368.

[21] Xunkai Li, Zhengyu Wu, Wentao Zhang, Henan Sun, Rong-Hua Li, and Guoren Wang. 2024. Adafgl: A new paradigm for federated node classification with topology heterogeneity. In *2024 IEEE 40th International Conference on Data Engineering*. IEEE, 2517–2530.

[22] Xunkai Li, Zhengyu Wu, Wentao Zhang, Yinlin Zhu, Rong-Hua Li, and Guoren Wang. 2023. FedGTA: Topology-Aware Averaging for Federated Graph Learning. *Proceedings of the VLDB Endowment* (2023), 41–50.

[23] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Proc. of AISTATS*. PMLR, 1273–1282.

[24] Xutong Mu, Yulong Shen, Ke Cheng, Xueli Geng, Jiaxuan Fu, Tao Zhang, and Zhiwei Zhang. 2023. Fedproc: Prototypical contrastive federated learning on non-iid data. *Future Generation Computer Systems* (2023), 93–104.

[25] Dinh C Nguyen, Quoc-Viet Pham, Pubudu N Pathirana, Ming Ding, Aruna Seneviratne, Zihuai Lin, Octavia Dobre, and Won-Joo Hwang. 2022. Federated learning for smart healthcare: A survey. *Comput. Surveys* (2022), 1–37.

[26] Hongbin Pei, Bingzhe Wei, Kevin Chen Chuan Chang, Yu Lei, and Bo Yang. 2020. Geom-gcn: Geometric graph convolutional networks. In *Proc. of ICLR*.

[27] Veličković Petar, Cucurull Guillem, Casanova Arantxa, Romero Adriana, Lio Pietro, and B Yoshua. 2018. Graph attention networks. In *Proc. of ICLR*.

[28] Oleg Platonov, Denis Kuznedelev, Michael Diskin, Artem Babenko, and Liudmila Prokhorenkova. 2023. A critical look at the evaluation of GNNs under heterophily: Are we really making progress?. In *Proc. of ICLR*.

[29] Liangqiong Qu, Yuyin Zhou, Paul Pu Liang, Yingda Xia, Feifei Wang, Ehsan Adeli, Li Fei-Fei, and Daniel Rubin. 2022. Rethinking architecture design for tackling data heterogeneity in federated learning. In *Proc. of CVPR*. 10061–10071.

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

[31] Yue Tan, Yixin Liu, Guodong Long, Jing Jiang, Qinghua Lu, and Chengqi Zhang. 2023. Federated learning on non-iid graphs via structural knowledge sharing. In *Proc. of AAAI*. 9953–9961.

[32] Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. 2022. Fedproto: Federated prototype learning across heterogeneous clients. In *Proc. of AAAI*. 8432–8440.

[33] Zihan Tan, Guancheng Wan, Wenke Huang, He Li, Guibin Zhang, Carl Yang, and Mang Ye. 2025. FedSPA: Generalizable Federated Graph Learning under Homophily Heterogeneity. In *Proc. of CVPR*.

[34] Guancheng Wan, Wenke Huang, and Mang Ye. 2024. Federated graph learning under domain shift with generalizable prototypes. In *Proc. of AAAI*. 15429–15437.

[35] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. 2019. Simplifying graph convolutional networks. In *Proc. of ICML*. Pmlr, 6861–6871.

[36] Zhenbang Xiao, Yu Wang, Shunyu Liu, Huiqiong Wang, Mingli Song, and Tongya Zheng. 2024. Simple graph condensation. In *Proc. of ECML*. Springer, 53–71.

[37] Han Xie, Jing Ma, Li Xiong, and Carl Yang. 2021. Federated graph classification over non-iid graphs. *Proc. of NeurIPS* (2021), 18839–18852.

[38] Yeyu Yan, Yinlin Zhu, Rong Hua Li, Boyang Pang, Zhengyu Wu, Guoren Wang, Guochen Yan, and Wentao Zhang. 2025. OpenFGL: A Comprehensive Benchmark for Federated Graph Learning. *Proceedings of the VLDB Endowment* (2025), 1305–1320.

[39] Xiyuan Yang, Wenke Huang, and Mang Ye. 2023. Dynamic personalized federated learning with adaptive differential privacy. *Proc. of NeurIPS* (2023), 72181–72192.

[40] Zhilin Yang, William Cohen, and Ruslan Salakhudinov. 2016. Revisiting semi-supervised learning with graph embeddings. In *Proc. of ICML*. PMLR, 40–48.

[41] Chao Zhang, Fangzhao Wu, Jingwei Yi, Derong Xu, Yang Yu, Jindong Wang, Yidong Wang, Tong Xu, Xing Xie, and Enhong Chen. 2023. Non-iid always bad? semi-supervised heterogeneous federated learning with local knowledge enhancement. In *Proc. of CIKM*. 3257–3267.

[42] Huanding Zhang, Tao Shen, Fei Wu, Mingyang Yin, Hongxia Yang, and Chao Wu. 2021. Federated graph learning–a position paper. *arXiv preprint arXiv:2105.11099* (2021).

[43] Jianqing Zhang, Yang Hua, Hao Wang, Tao Song, Zhengui Xue, Ruhui Ma, and Haibing Guan. 2023. Fedala: Adaptive local aggregation for personalized federated learning. In *Proc. of AAAI*. 11237–11244.

[44] Jianqing Zhang, Yang Liu, Yang Hua, and Jian Cao. 2024. Fedtgp: Trainable global prototypes with adaptive-margin-enhanced contrastive learning for data and model heterogeneity in federated learning. In *Proc. of AAAI*. 16768–16776.

[45] Ke Zhang, Carl Yang, Xiaoxiao Li, Lichao Sun, and Siu Ming Yiu. 2021. Subgraph federated learning with missing neighbor generation. *Proc. of NeurIPS* (2021), 6671–6682.

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

[47] Yinlin Zhu, Xunkai Li, Zhengyu Wu, Di Wu, Miao Hu, and Rong-Hua Li. 2024. FedTAD: topology-aware data-free knowledge distillation for subgraph federated learning. In *Proc. of IJCAI*. 5716–5724.

[48] Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. 2021. Data-free knowledge distillation for heterogeneous federated learning. In *Proc. of ICML*. PMLR, 12878–12889.