FREQUENCY-AWARE UNLEARNING FOR FEDERATED GRAPH CLUSTERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated graph unlearning (FGU) is a distributed learning paradigm that enables clients, motivated by privacy concerns or other considerations, to request the removal of specified graph data and the elimination of their influence from federated graph models. Existing methods have achieved significant progress, but generally assume graph labels are available, which is often difficult to satisfy in real-world scenarios. Under this limitation, most existing methods struggle to accurately characterize the impact of the data to be forgotten on cluster structures and clustering models, often resulting in incomplete unlearning and residual knowledge—a problem that remains largely underexplored. To address this issue, we propose **F**requency-**A**ware **U**nlearning for **Fed**erated Graph Clustering (FedFAU), which exploits frequency changes before and after unlearning to characterize the impact of forgotten data on local clustering structures and enable selective model updates. Specifically, we first partition each client graph into multiple blocks and perform local clustering, based on which block-level frequency information is constructed. Upon an unlearning request, we use the resulting frequency shifts to assess cluster reliability and selectively update affected global clusters, rather than retraining the entire global model. Extensive experiments on five benchmark datasets demonstrate the effectiveness and superiority of our method over competing approaches.

1 INTRODUCTION

With the widespread application of graph data in social networks (Fan et al. (2019)), recommendation systems (He et al. (2020)), and knowledge graphs (Dettmers et al. (2018)), Federated Graph Learning (FGL) has emerged as an important paradigm for collaborative graph learning under privacy constraints (Xie et al. (2021); Fu et al. (2022)). It enables clients to keep graph data locally and collaborate by exchanging only model information, thereby reducing the privacy risks of directly sharing raw data. However, the data involved in federated training are not always permanently available. Recent regulations and policies worldwide (Romandini et al. (2024)) grant users the right to request that entities holding their personal data stop using such data. Once users withdraw authorization, or nodes, edges, features, and even entire clients need to be removed, their influence may remain in historically trained models (Ai et al. (2026)).

Recently, Federated Graph Unlearning (FGU) has emerged as a promising solution, aiming to efficiently remove the influence of target data while preserving model performance on the remaining data (Chao et al. (2026)). Existing graph unlearning methods have achieved promising results on node-, edge-, and feature-level removal (Zhu et al. (2023); Liu & Fang (2024)). An essential prerequisite for most of these methods is the availability of label supervision, which provides direct prediction feedback for measuring the impact of unlearning. However, this prerequisite may not hold in practical scenarios, since labeled graph data are often difficult to obtain (Han et al. (2024); Song (2026)), especially for large-scale graphs. Once reliable supervision is unavailable, existing FGU methods can no longer directly measure the impact of unlearning through prediction errors. More importantly, graph nodes are strongly interconnected through graph topology. Removing a node, edge, or feature not only affects the target itself, but may also disturb neighboring representations and cluster structures. Therefore, under unsupervised settings, it remains challenging to identify which local structures and clustering results are actually affected by unlearning. Without accurate identification, insufficient updates may leave residual knowledge, while excessive updates may un-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

necessarily degrade unaffected regions. Notably, existing FGU frameworks still lack an effective mechanism for characterizing such clustering disturbances without ground-truth labels.

An intuitive approach is to directly compare clustering results before and after unlearning and regard the changed regions as affected ones. However, this strategy is unreliable in unsupervised federated graph clustering. On the one hand, without ground-truth labels, clustering results can only be represented by model-generated pseudo-labels or cluster assignments, whose changes may not accurately reflect structural disturbances. On the other hand, local data removal may propagate through graph topology and further affect neighboring representations and cluster structures. Therefore, clustering changes alone are insufficient to identify the affected regions accurately. Motivated by graph signal processing, in which low-frequency components generally capture local smoothness whereas high-frequency components characterize sharper variations (Min et al. (2020); Bo et al. (2021); Tan et al. (2024)), we regard clustering information as signals defined over graph topology and use frequency variations before and after unlearning to characterize local clustering disturbances. When unlearning changes the relationship between local topology and clustering assignments, the corresponding frequency representation also changes. This observation motivates us to use frequency variation as an additional criterion for identifying regions truly affected by unlearning, rather than relying solely on unavailable labels or unstable clustering results.

Based on these observations, we propose a novel frequency-aware unlearning method for federated graph clustering, which uses frequency variations before and after unlearning to characterize local clustering disturbances and selectively update the affected clustering knowledge. To this end, we design a block-wise frequency characterization and selective updating scheme. At the client, we first partition the local graph into multiple blocks and perform clustering based on node representations. We then combine local topology with clustering assignments to construct frequency representations for each block. When node-level, edge-level, or feature-level unlearning occurs, the original partition is retained, and the frequency information of the affected blocks is recomputed after removing the target data, allowing us to measure the resulting local disturbance. At the server, we align and aggregate clustering information uploaded from different clients to construct global clusters. Based on local frequency variations, we further evaluate the reliability of the corresponding clustering knowledge after unlearning. For significantly disturbed blocks, only the associated global clusters are selectively updated, while unaffected clusters are preserved. Meanwhile, the post-unlearning local update directions are used to calibrate the related model parameters. Our main contributions can be summarized as follows:

- **New research task.** To the best of our knowledge, we are the first to study federated graph unlearning in unsupervised graph clustering, with node, edge, and feature unlearning.
- **Novel proposed method.** We propose FedFAU, a frequency-aware federated graph unlearning framework that captures the residual influence of unlearned data via frequency variations and selectively removes its impact while preserving clustering performance.
- **Better results.** Extensive empirical results on five benchmark graph datasets validate the effectiveness and superiority of FedFAU compared with existing baseline methods.

2 Related Work

2.1 Federated Graph Learning

Federated Graph Learning (FGL) is a distributed learning paradigm that enables multiple clients to collaboratively train graph models through a central server without directly sharing their local graph data (Cai et al. (2024a;b); Wang et al. (2021; 2023; 2024; 2026); Li et al. (2024a;d); Meng et al. (2024)). Owing to its ability to balance data privacy and cross-client graph knowledge collaboration, FGL has attracted increasing attention and has been widely applied to various node-level graph learning tasks. For example, FedTAD (Zhu et al. (2024)) evaluates the reliability of local class knowledge from a topology-aware perspective and transfers reliable knowledge to the global model via data-free distillation for improved node classification. Moreover, FedGTA (Li et al. (2024b)) characterizes client-specific graph distributions using topology-aware local information and performs personalized aggregation accordingly. Similarly, FedNCN (Liu et al. (2025)) exploits local clustering signals to repair cross-subgraph links disrupted by graph partitioning and learns consensus prototypes for unlabeled federated graph clustering. Commonly, these methods focus on acquiring

and sharing graph knowledge during conventional training. However, when specific graph data are requested to be removed, their historical influence may remain in the federated model. To this end, we perform a remove-and-replace update on the global prototypes and model parameters, eliminating the historical influence of specified graph data while maintaining clustering performance.

2.2 Federated Graph Unlearning

Federated Graph Unlearning (FGU) aims to eliminate the influence of specified graph data from federated graph models while preserving the knowledge learned from the remaining data (Wu et al. (2023); Li et al. (2024c); Sinha et al. (2025)). As machine unlearning has been extended to graph-structured data, recent studies have explored different forms of graph information removal. For example, FedLU (Zhu et al. (2023)) targets heterogeneous federated knowledge graphs and employs knowledge distillation to forget specified knowledge while coordinating local and global knowledge updates. FedDM (Liu & Fang (2024)) further leverages diffusion models to facilitate knowledge removal in federated knowledge graphs while preserving the remaining knowledge. More recently, PAGE (Ai et al. (2026)) studies node, edge, and feature unlearning in federated graph learning and coordinates local unlearning with global knowledge removal to reduce the residual influence of forgotten graph information during federated collaboration. Although existing graph unlearning methods have achieved promising progress in removing designated graph knowledge, they are mainly developed for supervised graph learning or knowledge graph representation learning. Directly applying them to unlabeled federated graph clustering makes it difficult to accurately characterize how forgotten data alter the underlying clustering structures. In contrast, we characterize the impact of unlearning through changes in local clustering structures before and after unlearning, and accordingly perform selective updates on the affected global clustering knowledge.

3 Methodology

Fig. 1 shows the architecture of FedFAU. The core idea is to characterize the impact of unlearning based on the changes in frequency information before and after unlearning, thereby enabling precise unlearning while preserving model performance in an unlabeled setting. This process mainly consists of two stages: blockwise frequency extraction and frequency-guided graph unlearning.

3.1 Notations

In the federated graph clustering setting, a global graph $\mathcal{G}$ with K clusters is partitioned into M subgraphs, which are distributed across M clients. For simplicity, we take a single client as an example, whose local subgraph is denoted as $\bar{\mathcal{G}} = (\bar{\mathcal{V}}, \bar{\mathcal{E}})$ with $\bar{K}$ clusters, where $\bar{\mathcal{V}}$ and $\bar{\mathcal{E}}$ denote the node set and edge set, respectively. Suppose that $\bar{\mathcal{G}}$ contains the node attribute matrix $\bar{\mathbf{X}} \in \mathbb{R}^{\bar{N} \times d}$ and the original adjacency matrix $\hat{\mathbf{A}} \in \mathbb{R}^{\bar{N} \times \bar{N}}$, where $\bar{N}$ and d denote the number of nodes and the node attribute dimension in $\bar{\mathcal{G}}$. Let $\widetilde{\mathbf{D}} \in \mathbb{R}^{\bar{N} \times \bar{N}}$ be the original degree matrix. The normalized adjacency matrix $\tilde{\mathbf{A}} \in \mathbb{R}^{\bar{N} \times \bar{N}}$ is then computed as $\tilde{\mathbf{A}} = \tilde{\mathbf{D}}^{-1/2}(\hat{\mathbf{A}} + \mathbf{I})\tilde{\mathbf{D}}^{-1/2}$, where $\mathbf{I} \in \mathbb{R}^{\bar{N} \times \bar{N}}$ denotes the identity matrix. For clarity, the main notations used throughout this paper are summarized in Appendix A.1.

3.2 Blockwise Frequency Extraction

This section focuses on blockwise frequency extraction for a single subgraph, aiming to extract the frequency information of each block. Specifically, we decompose the extraction process into four steps to progressively obtain blockwise frequency information. In the first step, we partition each local subgraph into B structural blocks via random walks, and B is the parameter that is analyzed in the Impact of Hyper-parameter. The partitioning process is formulated as

$$P(v_t = v \mid v_{t-1} = u) = \begin{cases} \dfrac{1}{\tilde{d}_u}, & (u, v) \in \bar{\mathcal{E}}, \\ 0, & \text{otherwise}, \end{cases} \tag{1}$$

where $u, v \in \bar{\mathcal{V}}$ denote two different nodes in $\bar{\mathcal{G}}$, v_t denotes the node visited at step t, $\tilde{d}_u \in \tilde{\mathbf{D}}$ denotes the degree of node u. In this way, $\bar{\mathcal{G}}$ is partitioned into $\hat{\mathcal{G}} = (\bar{\mathcal{V}}, \hat{\mathcal{E}})$, where the edge set is

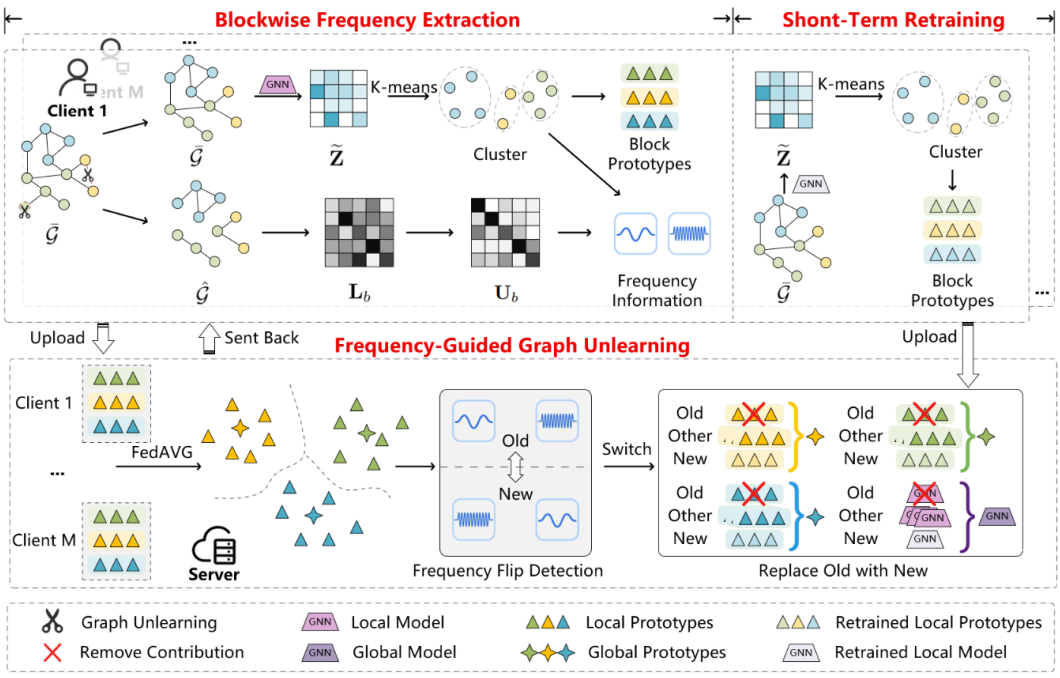

Figure 1: FedFAU consists of two components: blockwise frequency extraction and frequency-guided graph unlearning. The former captures block-level frequency information to characterize clustering changes induced by unlearning, while the latter identifies affected blocks and selectively updates their clustering knowledge. Together, they form a federated graph unlearning framework.

represented by partitioned adjacency matrix $\bar{\mathbf{A}} \in \mathbb{R}^{\bar{N}\times\bar{N}}$ and partitioned degree matrix $\bar{\mathbf{D}} \in \mathbb{R}^{\bar{N}\times\bar{N}}$. Specifically, for the b-th block, its adjacency matrix and degree matrix are denoted by $\bar{\mathbf{A}}_b \in \mathbb{R}^{n_b\times n_b}$ and $\bar{\mathbf{D}}_b \in \mathbb{R}^{n_b\times n_b}$, respectively. Here, n_b is the number of nodes in the b-th block.

In the second step, we first employ a GNN to obtain the node representations of the subgraph as $\bar{\mathbf{Z}}^{(l)} = \sigma\left(\tilde{\mathbf{A}}\bar{\mathbf{Z}}^{(l-1)}\bar{\mathbf{W}}^{(l)}\right)$, where $\bar{\mathbf{Z}}^{(l)} \in \mathbb{R}^{\bar{N}\times d'}$ denotes the node embedding of the l-th encoder layer, and $\bar{\mathbf{Z}}^{(0)} = \bar{\mathbf{X}}$. $\bar{\mathbf{W}}^{(l)}$ represents the learnable weight matrix at the l-th layer, and $\sigma(\cdot)$ is the activation function. The final embeddings are denoted by $\tilde{\mathbf{Z}} \in \mathbb{R}^{\bar{N}\times d''}$. Then, we employ K-means to obtain local prototype $\bar{\mathbf{C}} \in \mathbb{R}^{\bar{K}\times d''}$, and further compute the soft assignment matrix $\bar{\mathbf{Q}} \in \mathbb{R}^{\bar{N}\times\bar{K}}$ and the node count matrix $\bar{\mathbf{S}} \in \mathbb{R}^{B\times\bar{K}}$, where $\bar{s}_{b,k}$ represents the number of nodes in the k-th cluster of the b-th block. Taking the b-th block as an example, its local prototypes are denoted by $\bar{\mathbf{C}}_b \in \mathbb{R}^{\bar{K}\times d''}$, from which the corresponding soft assignment matrix $\bar{\mathbf{Q}}_b \in \mathbb{R}^{n_b\times\bar{K}}$ is obtained.

In the third step, inspired by FedSSP (Tan et al. (2024)), we treat $\bar{\mathbf{Q}}$ as a graph signal and combine it with the topology of $\bar{\mathcal{G}}$ to extract frequency information that characterizes variations in cluster information. For the b-th block, we first construct its graph Laplacian matrix $\mathbf{L}_b \in \mathbb{R}^{n_b\times n_b}$ as $\mathbf{L}_b = \mathbf{I} - \bar{\mathbf{D}}_b^{-\frac{1}{2}}\bar{\mathbf{A}}_b\bar{\mathbf{D}}_b^{-\frac{1}{2}}$. Moreover, we perform eigendecomposition on $\mathbf{L}_b$ as

$$\mathbf{L}_b = \mathbf{U}_b\mathbf{\Lambda}_b\mathbf{U}_b^\top, \qquad \mathbf{\Lambda}_b = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_{n_b}), \qquad 0 \le \lambda_1 \le \lambda_2 \le \cdots \le \lambda_{n_b}. \tag{2}$$

where $\mathbf{U}_b \in \mathbb{R}^{n_b\times n_b}$ and $\mathbf{\Lambda}_b \in \mathbb{R}^{n_b\times n_b}$ denote the eigenvectors and eigenvalues of $\mathbf{L}_b$, respectively. Then, we apply the Graph Fourier Transform to obtain the graph Fourier coefficient matrix $\hat{\mathbf{Q}}_b \in \mathbb{R}^{n_b\times\bar{K}}$ as $\hat{\mathbf{Q}}_b = \mathbf{U}_b^\top\bar{\mathbf{Q}}_b$. We further classify components with $\lambda \le 0.5$ as low-frequency information and the rest as high-frequency information. Accordingly, the cluster-frequency information of the b-th block is computed as

$$\left(r_{b,k}^L, r_{b,k}^H\right) = \left(\frac{\sum_{\lambda_l\le 0.5}\left|\hat{q}_b^{k,l}\right|^2}{\sum_{l=1}^{n_b}\left|\hat{q}_b^{k,l}\right|^2}, \frac{\sum_{\lambda_l>0.5}\left|\hat{q}_b^{k,l}\right|^2}{\sum_{l=1}^{n_b}\left|\hat{q}_b^{k,l}\right|^2}\right), \tag{3}$$

where $r_{b,k}^L \in \mathbf{R}^L$ and $r_{b,k}^H \in \mathbf{R}^H$ denote the proportions of low- and high-frequency information of the k-th cluster in the b-th block. Here, $\mathbf{R}^L \in \mathbb{R}^{B\times\bar{K}}$ and $\mathbf{R}^H \in \mathbb{R}^{B\times\bar{K}}$ denote low- and high-frequency information matrices. A larger r_b^L indicates a more consistent local cluster distribution, whereas a larger r_b^H reflects more pronounced variations in local cluster distributions.

Furthermore, to preserve local graph information and learn clustering-friendly representations, we minimize the objective $\mathcal{L} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{gra}} + \mathcal{L}_{\text{dec}}$, where $\mathcal{L}_{\text{rec}}$, $\mathcal{L}_{\text{gra}}$ and $\mathcal{L}_{\text{dec}}$ are calculated as

$$\mathcal{L}_{\text{rec}} = \frac{1}{\bar{N}} \sum_{u\in\bar{\mathcal{V}}} \left(1 - \frac{\bar{x}_u^\top \hat{x}_u}{\|\bar{x}_u\|_2 \|\hat{x}_u\|_2}\right), \tag{4}$$

$$\mathcal{L}_{\text{gra}} = \frac{1}{2}\left[\frac{1}{|E^+|}\sum_{(u,v)\in E^+} \text{softplus}(-h_{uv}) + \frac{1}{|E^-|}\sum_{(u,v)\in E^-} \text{softplus}(h_{uv})\right], h_{uv} = 2\bar{\mathbf{z}}_u^\top \bar{\mathbf{z}}_v, \tag{5}$$

$$\mathcal{L}_{\text{dec}} = \frac{1}{\bar{N}} \sum_{u=1}^{\bar{N}} \sum_{j=1}^{\bar{K}} \bar{p}_{uj} \log \frac{\bar{p}_{uj}}{\bar{q}_{uj}}, \tag{6}$$

where h_{uv} represents the similarity between u and v, $\hat{x}_u \in \hat{\mathbf{X}}$ is the reconstructed attribute vector of u, and $\hat{\mathbf{X}} \in \mathbb{R}^{\bar{N}\times d}$ denotes the reconstructed attribute matrix, E^+ and E^- denote the sets of pairs of connected and unconnected nodes, respectively. Additionally, $\bar{q}_{uj} \in \bar{\mathbf{Q}}$ denotes the soft assignment probability of node u in the j-th cluster, $\bar{p}_{uj} \in \bar{\mathbf{P}}$ denotes the sharpened target probability of node u in the j-th cluster, and the target distribution $\bar{\mathbf{P}} \in \mathbb{R}^{\bar{N}\times\bar{K}}$ is obtained by sharpening $\bar{\mathbf{Q}}$.

In the last step, we upload the model parameters $\bar{\theta}$, a set of local prototypes $\{\bar{\mathbf{C}}_b\}_{b=1}^B$, the node count matrix $\bar{\mathbf{S}}$, low-frequency information matrix $\mathbf{R}^L$ and high-frequency information matrix $\mathbf{R}^H$.

Moreover, when a frequency flip is detected after unlearning, we perform short-term local retraining to maintain model performance. The retraining reuses all normal training steps except graph partitioning and frequency computation.

In summary, the proposed blockwise frequency extraction offers two advantages: 1) it extracts and uploads frequency information to characterize changes in cluster structure; and 2) frequency information serves as an unlearning signal for subsequent frequency-aware federated graph unlearning.

3.3 Frequency-Guided Graph Unlearning

After receiving information uploaded by multiple clients, the server identifies structural blocks affected by unlearning, enabling precise unlearning while preserving model performance.

Firstly, we use the Hungarian algorithm to establish cluster correspondences among the uploaded local prototypes across clients. Then, we aggregate the local prototypes, the cluster-size vector, and model parameters via FedAvg to obtain the global prototypes $\hat{\mathbf{C}} \in \mathbb{R}^{K\times d''}$, the global cluster node-count vector $\hat{S} \in \mathbb{R}^K$, and the global model parameters at the t-th communication round $\theta_g^{(t)}$.

Next, for the k-th cluster in the b-th block, we calculate the difference between its low- and high-frequency information as $\eta_{b,k} = r_{b,k}^L - r_{b,k}^H$. Upon receiving an unlearning request, we compare $\eta_{b,k}$ before and after unlearning to determine whether a frequency flip occurs, as

$$\zeta_{b,k} = \text{Flip}(\eta_{b,k}^{\text{old}}, \eta_{b,k}^{\text{new}}) = \begin{cases} 1, & \text{if } \eta_{b,k}^{\text{old}} \eta_{b,k}^{\text{new}} < 0, \\ 0, & \text{otherwise}, \end{cases} \tag{7}$$

where Flip$(\cdot,\cdot)$ is the frequency-flip indicator function, $\eta_{b,k}^{old}$ and $\eta_{b,k}^{new}$ represent the difference between low- and high-frequency information of the k-th cluster in the b-th block before and after the unlearning, respectively. Subsequently, if $\zeta_{b,k} = 1$, we remove the pre-unlearning contribution of the

corresponding local cluster from the global cluster and incorporate its post-unlearning contribution, updating the k-th global prototype $\hat{c}_k \in \hat{\mathbf{C}}$ as

$$\hat{c}_k = \frac{\hat{S}_k \hat{c}_k + \bar{s}_{b,k}^{\text{new}} \bar{c}_{b,k}^{\text{new}} - \bar{s}_{b,k}^{\text{old}} \bar{c}_{b,k}^{\text{old}}}{\hat{S}_k + \bar{s}_{b,k}^{\text{new}} - \bar{s}_{b,k}^{\text{old}}}, \tag{8}$$

where $\hat{S}_k \in \hat{S}$ denotes the total number of nodes in the k-th global cluster before unlearning, $\hat{c}_k \in \hat{\mathbf{C}}$ denotes the k-th global prototype, and $\bar{c}_{b,k}^{\text{old}}$ and $\bar{c}_{b,k}^{\text{new}}$ are the corresponding local cluster prototypes before and after unlearning. Here, $\bar{s}_{b,k}^{\text{old}}$ and $\bar{s}_{b,k}^{\text{new}}$ are the node counts for the k-th cluster in the b-th block before and after unlearning, respectively. In addition, to eliminate the residual influence of the data requested for removal from the global model, inspired by FedEraser (Liu et al. (2021)), we calibrate the global model parameters by removing the pre-unlearning update contribution and incorporating the update direction obtained from short-term retraining on the remaining data as

$$\theta_g^{(t)} \leftarrow \theta_g^{(t-1)} + \alpha \left(\left\| \Delta_t^{\text{old}} \right\| \frac{\hat{\Delta}_t^{\text{new}}}{\left\| \hat{\Delta}_t^{\text{new}} \right\|} - \Delta_t^{\text{old}} \right), \tag{9}$$

where $\Delta_t^{\text{old}} = \bar{\theta}^t - \theta_g^{t-1}$ denotes the local update before unlearning, while $\hat{\Delta}_t^{\text{new}} = \theta_{\text{ul}}^t - \theta_g^{t-1}$ denotes the update direction derived from short-term re-training on the post-unlearning graph. Here, θ_g^{t-1} is the global model distributed at round $t-1$, $\bar{\theta}^t$ is the locally trained model before unlearning, θ_{ul}^t is obtained after a few epochs of short-term training on the post-unlearning graph, and α denotes the corresponding FedAvg aggregation weight.

Finally, the server distributes updated global prototypes $\hat{\mathbf{C}}$ and model parameters $\theta_g^{(t)}$ to clients for local learning. Here, $\hat{\mathbf{C}}$ updates local cluster centers, while $\theta_g^{(t)}$ initializes local models with reduced residual influence from the unlearned data. This global-local interaction enables effective unlearning while preserving clustering performance. We provide the algorithm in Appendix B.2.

The merits of frequency-guided graph unlearning can be summarized as follows: 1) it identifies blocks affected by unlearning through frequency-domain changes; and 2) it performs remove-and-replace updates on global prototypes and model parameters to mitigate the impact of unlearning.

4 Experiments

4.1 Experiment Setup.

Benchmark Datasets. Following the experimental protocol of FedNCN (Liu et al. (2025)), each original graph is partitioned into multiple client-specific subgraphs to simulate a federated graph setting. We consider three federation scales with 5, 10, and 20 clients, respectively, where each client holds only a local subgraph of the original complete graph. We conduct experiments on five benchmark graph datasets, including CiteSeer (Liu et al. (2023a)), Cora (Platonov et al. (2023)), Amazon-Computer (Am.Co for abbreviation), Amazon-Photo (Am.Ph for abbreviation) (Lin et al. (2021)) , and PubMed (Jiang et al. (2024)). Detailed descriptions are provided in Appendix B.1.

Baselines. We list Retrain and compare FedFAU with six representative supervised unlearning methods, including three centralized methods: D2DGN (Sinha et al. (2025)), MEGU (Li et al. (2024c)), GIF (Wu et al. (2023)), and three distributed methods: FedDM (Liu & Fang (2024)), FedLU (Zhu et al. (2023)), and PAGE (Ai et al. (2026)). Details are provided in Appendix B.3.

Implementation Details. To ensure experimental fairness, all methods are evaluated under identical hardware and settings. For all experiments, we employ a two-layer GNN encoder to learn node representations, with hidden dimensions of 128 and 32, respectively. We adopt FedAdam for federated training with a learning rate of 1e-3. The model is trained for 20 communication rounds with 10 local epochs per round, while 100 rounds are used for PubMed and Am.Ph with the same local epochs. In FedFAU, 2 epochs of short-term retraining are performed after an unlearning request to estimate the post-unlearning update direction. For unlearning settings, we randomly remove 10% of nodes, edges, and features. Specifically, in the "Comparison with Supervised FGL Method" section,

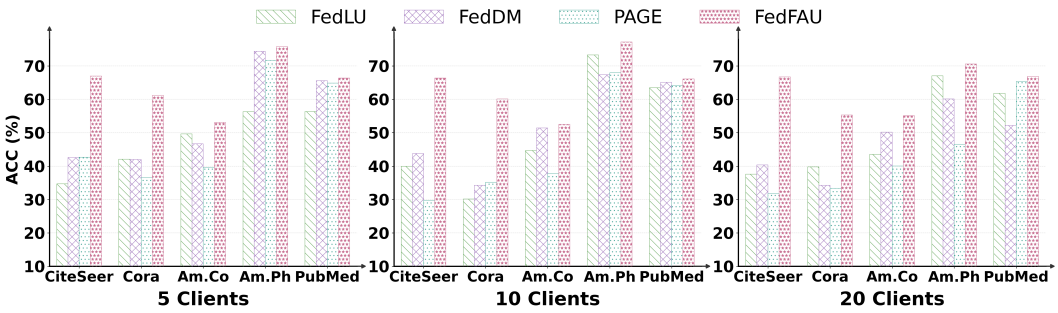

Figure 2: Performance comparison of FedFAU with three distributed unlearning methods on five graph datasets.

we randomly remove 10% of nodes. All experiments are implemented with PyTorch 2.11.0 and a single NVIDIA GeForce RTX 3090 GPU.

Unlearning Evaluation. To assess the influence of forgotten data after unlearning, we evaluate forgetting using backdoor-based Attack Success Rate (ASR) (Wu et al. (2022); Liu & Liu (2025)). We randomly select a client as the attacked client and a cluster as the target cluster. Among the nodes to be forgotten, 30% are selected for poisoning by adding a trigger and encouraging them to be assigned to the target cluster. During unlearning, FedFAU and Retrain remove the same forgotten nodes without backdoor optimization. For evaluation, we fix up to 100 non-target nodes from the attacked client, attach the same trigger, and compute ASR as the proportion assigned to the target cluster, where lower values indicate better forgetting. All experiments employ the same settings.

Clustering Metrics. To evaluate the performance of FedFAU and all baseline methods, we adopt four standard metrics, including Accuracy (ACC)(Pan et al. (2018); Wang et al. (2019); Bo et al. (2020)), Normalized Mutual Information (NMI) (Tu et al. (2021); Gong et al. (2022); Liu et al. (2022)), Adjusted Rand Index (ARI) (Yang et al. (2023); Liu et al. (2023b)), and F1 Score (F1) (Tu et al. (2024); Wen et al. (2024)).

4.2 Experimental Results

Clustering Performance Comparison. To validate the effectiveness of FedFAU, which is the first framework for graph unlearning in unlabeled federated graph clustering, we adopt six supervised graph unlearning methods and extend them into unsupervised versions to ensure a fair comparison. To mitigate the impact of randomness, each experiment is conducted five times, and the mean and standard deviation of the four clustering metrics are reported in Table 1. From the experimental results, we observe that FedFAU consistently outperforms competing methods across five datasets under different client partition settings and achieves performance close to full retraining. This can be attributed to two aspects. First, existing methods are mainly designed for supervised graph unlearning and rely on label information to guide the removal of unlearned data, which limits their effectiveness in unlabeled federated graph clustering. Second, FedFAU exploits blockwise frequency information to capture the influence of unlearned data and selectively updates the affected clustering knowledge, thereby achieving effective unlearning while preserving clustering performance.

Unlearning Performance Comparison. To evaluate the unlearning effectiveness of FedFAU, we compare it with six graph unlearning methods under the same experimental settings. As shown in Table 2, FedFAU consistently outperforms competing methods across five datasets under different client partition settings, achieving performance comparable to Retrain. This advantage can be attributed to two aspects. First, existing methods mainly focus on supervised graph unlearning and rely on label information to guide target knowledge removal, limiting their effectiveness in unlabeled scenarios. Second, FedFAU exploits block-frequency information to characterize the impact of unlearned data, identify affected structural blocks, and achieve selective forgetting via removing outdated knowledge and incorporating updated knowledge.

Comparison with Supervised Unlearning Methods. To further verify the effectiveness of FedFAU, we compare it with three supervised distributed unlearning methods: FedLU, FedDM, and PAGE, where each compared method is trained using only 2% labeled data. As shown in Fig. 2, the

Table 1: Clustering performance comparison of FGU methods under unsupervised settings. All methods are evaluated under the same setting on five benchmark datasets for fair comparison.

Methods	CiteSeer (Client 5)				CiteSeer (Client 10)				CiteSeer (Client 20)			
	ACC	NMI	ARI	F1	ACC	NMI	ARI	F1	ACC	NMI	ARI	F1
Retrain	67.8±0.7	44.2±0.6	44.2±0.3	62.7±0.7	66.5±1.0	43.2±0.9	43.6±0.7	60.8±1.0	68.8±0.2	45.5±0.3	45.4±0.4	63.1±0.2
D2DGN†	39.0±3.0	9.8±2.5	7.7±5.2	23.2±3.1	40.7±2.9	14.3±1.7	10.2±2.8	27.6±1.0	40.0±1.5	16.6±1.6	7.9±1.2	29.9±1.9
MEGU†	57.7±2.9	16.7±1.0	22.8±2.7	22.8±1.0	53.8±3.5	8.2±2.1	4.0±4.7	18.6±1.3	60.3±3.5	5.7±4.0	4.0±5.4	19.7±4.2
GIF†	42.5±4.8	14.2±2.9	15.3±4.4	27.4±2.2	41.1±2.7	14.9±2.1	11.5±2.7	27.5±1.2	41.6±2.0	17.3±1.4	8.9±0.8	30.8±0.9
FedLU*	34.7±1.1	8.2±1.0	6.9±1.3	23.1±0.8	37.7±2.1	10.8±0.8	7.4±1.7	25.2±1.0	41.6±1.9	16.5±1.5	8.4±0.9	30.2±1.5
FedDM*	40.6±1.7	10.4±1.3	11.7±2.5	25.4±0.7	39.8±3.9	11.8±1.3	8.4±3.1	26.3±1.8	43.1±1.4	17.0±1.0	8.7±1.2	31.0±1.0
PAGE*	30.2±0.7	4.5±0.6	0.4±0.4	10.9±1.2	30.6±0.9	4.6±0.3	-0.9±0.2	9.9±0.8	32.5±0.8	8.5±0.4	-0.0±0.3	10.2±0.6
Ours	**66.7±0.3**	**43.2±0.2**	**43.6±0.5**	**61.6±0.3**	**66.3±0.1**	**43.2±0.1**	**43.8±0.1**	**60.8±0.1**	**66.5±0.2**	**43.8±0.2**	**44.3±0.3**	**60.8±0.1**
Methods	**Cora (Client 5)**				**Cora (Client 10)**				**Cora (Client 20)**			
Retrain	59.3±2.8	34.5±0.7	31.0±2.4	57.7±3.1	60.3±0.4	34.4±0.3	31.7±0.7	58.6±0.3	55.9±0.9	33.4±0.3	28.6±0.7	54.2±0.7
D2DGN†	35.7±2.2	15.2±1.4	11.9±1.6	26.1±1.5	33.3±1.5	12.1±1.2	6.8±1.3	22.5±1.1	35.8±1.4	15.1±1.2	6.0±1.2	24.4±0.8
MEGU†	36.8±2.6	14.9±0.9	8.8±3.2	23.0±1.8	45.0±4.7	11.3±1.7	5.8±1.8	20.3±2.4	50.6±3.9	9.7±1.6	1.8±1.3	17.8±1.6
GIF†	36.8±3.3	15.5±1.5	12.3±2.6	26.8±1.5	33.5±2.2	11.7±1.1	6.7±1.5	23.2±1.8	35.9±1.3	14.8±1.0	5.8±0.8	24.2±1.2
FedLU*	27.9±0.9	4.4±0.4	1.8±0.2	17.8±0.4	29.7±0.6	7.2±0.5	1.8±0.4	18.4±0.5	33.3±0.9	10.5±0.4	1.8±0.8	20.3±0.9
FedDM*	41.1±3.2	18.0±1.0	16.5±2.2	27.3±1.7	34.6±1.1	10.8±0.6	6.0±1.0	21.8±1.0	35.5±1.5	13.0±0.7	3.3±1.0	22.7±0.7
PAGE*	33.4±1.6	14.8±1.7	9.7±1.7	11.0±3.3	34.4±0.8	16.9±0.6	8.6±0.9	11.1±1.5	34.2±1.3	18.5±0.8	7.3±1.0	10.4±2.5
Ours	**61.1±0.1**	**35.2±0.2**	**32.8±0.2**	**59.4±0.1**	**60.6±0.1**	**34.8±0.1**	**32.1±0.1**	**58.9±0.1**	**55.2±0.1**	**33.1±0.0**	**28.1±0.1**	**53.7±0.1**
Methods	**Am.Co (Client 5)**				**Am.Co (Client 10)**				**Am.Co (Client 20)**			
Retrain	53.0±1.7	52.9±1.1	36.7±1.0	42.2±4.1	51.3±0.5	52.0±0.5	34.7±0.4	38.7±0.7	51.1±0.1	52.9±0.3	34.6±0.3	40.0±1.2
D2DGN†	43.3±3.1	13.4±3.5	10.3±4.5	20.9±1.6	46.0±1.7	13.9±1.9	10.4±2.1	22.9±0.8	46.2±1.8	11.9±1.0	8.2±0.8	26.1±1.3
MEGU†	51.5±1.0	1.0±0.2	2.5±0.3	11.8±0.5	47.7±3.1	2.9±0.4	1.7±1.4	15.7±0.3	45.5±4.1	3.1±0.1	1.0±0.4	19.3±0.8
GIF†	51.6±3.9	21.4±2.5	17.5±2.7	25.8±2.6	52.4±4.8	19.7±1.5	15.3±3.1	25.8±1.3	53.6±2.2	15.6±1.1	12.9±1.5	29.6±1.8
FedLU*	37.0±2.2	10.0±1.8	5.2±1.6	18.9±1.4	44.8±0.9	11.4±1.7	7.1±1.7	21.2±0.6	44.9±2.9	10.5±0.8	7.1±0.7	24.7±1.4
FedDM*	49.8±3.6	16.2±3.7	12.9±5.0	23.2±2.0	52.3±4.8	16.1±2.2	13.3±3.3	25.0±2.2	54.6±3.9	14.1±1.8	11.7±2.9	28.7±2.1
PAGE*	21.4±0.6	1.5±0.1	0.1±0.1	4.5±0.8	21.4±0.4	2.4±0.0	-0.1±0.1	4.0±0.2	22.1±0.6	2.7±0.0	-0.1±0.0	3.5±0.4
Ours	**53.0±0.1**	**51.9±0.1**	**35.3±0.1**	**43.6±0.1**	**52.5±0.0**	**54.1±0.1**	**37.6±0.1**	**41.3±0.0**	**55.1±0.0**	**52.8±0.0**	**35.1±0.1**	**41.1±0.0**
Methods	**Am.Ph (Client 5)**				**Am.Ph (Client 10)**				**Am.Ph (Client 20)**			
Retrain	74.4±2.6	64.8±2.7	55.1±2.5	67.8±4.4	73.1±2.5	64.4±2.7	54.8±2.7	66.1±3.4	67.8±4.2	60.4±4.3	47.3±5.7	62.4±3.9
D2DGN†	67.1±5.2	28.0±2.3	31.7±6.2	24.5±1.9	67.1±4.3	13.5±4.0	11.5±4.0	20.7±2.1	66.2±2.8	12.1±1.6	9.7±1.7	33.2±1.6
MEGU†	56.6±1.7	1.3±0.2	1.4±0.2	12.9±0.5	57.9±4.8	2.2±0.2	2.9±0.8	14.2±0.9	61.1±5.2	3.1±0.3	2.5±1.2	23.0±1.2
GIF†	60.8±2.2	35.1±2.7	33.6±3.3	28.1±3.0	58.6±4.3	21.5±3.3	18.7±5.6	26.3±1.6	57.9±3.0	12.7±1.1	8.4±1.4	33.1±1.2
FedLU*	55.4±4.2	23.5±3.0	20.4±3.0	23.3±1.2	59.4±7.3	18.3±3.7	17.1±6.7	24.6±1.6	61.4±3.9	12.5±1.0	10.3±1.4	32.1±2.2
FedDM*	60.4±4.1	25.5±4.3	27.2±6.5	27.2±3.1	55.3±2.7	17.7±1.6	15.0±1.8	24.3±2.0	59.4±2.2	11.2±1.1	8.3±1.6	33.2±1.4
PAGE*	26.2±1.2	8.6±1.8	4.3±0.9	5.8±1.1	24.7±0.5	3.0±0.1	0.4±0.1	4.8±0.8	25.4±0.5	3.6±0.1	0.3±0.2	4.5±1.0
Ours	**72.3±3.5**	**62.0±3.7**	**52.2±4.0**	**64.8±5.4**	**74.7±2.5**	**65.2±2.4**	**55.7±2.3**	**67.8±3.6**	**66.9±3.7**	**61.6±3.2**	**47.8±3.8**	**63.1±4.9**
Methods	**PubMed (Client 5)**				**PubMed (Client 10)**				**PubMed (Client 20)**			
Retrain	62.7±3.8	25.2±2.5	23.1±4.1	62.7±4.4	62.3±3.8	24.5±2.1	22.3±4.0	62.3±4.4	63.0±4.0	25.4±2.5	23.3±4.4	62.8±4.7
D2DGN†	55.2±2.9	13.9±2.6	10.9±3.5	44.9±3.6	61.0±2.9	10.5±2.5	12.5±2.3	45.0±2.3	58.5±3.8	7.7±3.4	9.3±3.0	39.9±3.3
MEGU†	60.2±5.0	17.0±1.7	14.4±3.4	44.1±3.1	61.1±3.9	1.0±0.6	0.9±1.1	33.0±1.1	62.4±3.8	1.1±0.5	-0.6±0.8	31.8±0.7
GIF†	54.6±3.9	14.5±3.5	11.6±4.0	44.8±5.0	59.4±2.1	10.0±2.4	11.3±2.4	43.7±2.3	59.2±3.5	7.9±3.4	9.5±3.0	40.4±2.7
FedLU*	55.1±3.0	3.2±1.5	5.9±1.7	37.4±1.0	54.8±4.2	3.2±1.2	5.3±2.2	36.7±1.9	57.4±4.8	3.2±0.9	5.5±2.1	36.8±1.9
FedDM*	54.3±3.9	9.0±4.8	8.9±3.3	41.6±3.7	57.2±3.8	6.5±2.8	6.8±2.7	40.0±3.5	54.8±4.0	4.3±2.1	4.6±2.5	36.6±2.9
PAGE*	52.8±2.8	9.2±1.3	5.8±3.3	25.8±3.6	50.0±2.0	9.6±2.2	6.3±3.0	26.5±3.5	56.1±1.5	11.1±3.0	9.7±3.0	26.7±3.2
Ours	**62.6±3.8**	**24.6±2.3**	**22.6±4.1**	**62.6±4.4**	**62.4±3.7**	**24.5±2.1**	**22.4±3.8**	**62.4±4.3**	**63.0±3.9**	**25.5±2.4**	**23.5±4.2**	**62.9±4.6**

∗ Note that these supervised FGU methods are adapted to the unsupervised scenario.
† Note that these supervised graph-centralized unlearning methods are converted to distributed methods.

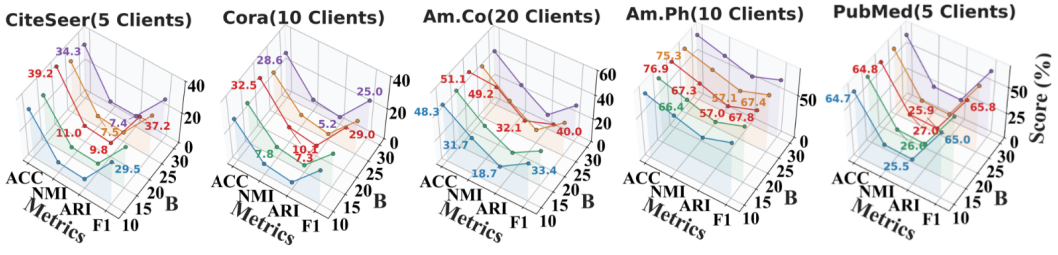

Figure 3: Effect of B on the performance of FedFAU across the five datasets.

experimental results demonstrate that FedFAU consistently achieves superior ACC across different datasets and client partitions, highlighting its effectiveness in node unlearning scenarios. Three other centralized unlearning methods are compared in Appendix B.4

Table 2: Unlearning performance comparison of FGU methods .

Method	PubMed (5 Clients)	Cora (10 Clients)	Am.Ph (10 Clients)	CiteSeer (20 Clients)	Am.Co (20 Clients)
Retrain	0.0	9.0	0.0	0.0	0.0
D2DGN	100.0	99.0	100.0	90.8	0.0
FedLU	5.0	100.0	0.0	0.0	100.0
FedDM	100.0	63.0	100.0	89.5	1.0
MEGU	92.0	31.0	2.0	0.0	0.0
PAGE	16.0	35.0	93.0	100.0	20.0
GIF	100.0	100.0	100.0	0.0	0.0
FedFAU	**0.0**	**14.0**	**0.0**	0.0	**0.0**

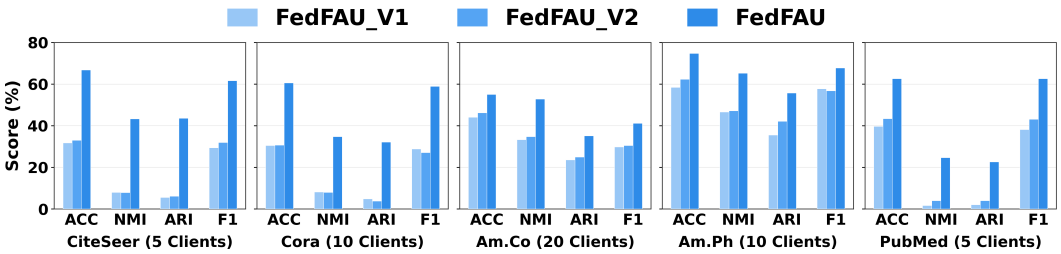

Figure 4: Ablation studies for the Block-Frequency component.

Effect of Hyper-Parameter B. FedFAU introduces a hyper-parameter B to control the number of local blocks. We vary B from 10 to 30 with a step size of 5 on five datasets. As shown in Fig. 3, we can observe that: 1) B plays an important role in FedFAU, with clustering performance generally improving first and then declining as B increases; 2) a small B yields coarse blocks that may obscure local frequency variations, limiting the characterization of clustering changes; 3) a large B over-fragments local structures, making the extracted frequency information less stable and potentially introducing noise. In particular, $B = 20$ achieves the best overall performance across the five datasets. Therefore, we set $B = 20$ as the default value in the following experiments.

Module Ablation. To validate the effectiveness of the proposed Block-Frequency component, we compare FedFAU with its variants on five graph datasets. Concretely, "FedFAU_V1" denotes the variant with both block partition and frequency information removed, and "FedFAU_V2" retains block partition but removes frequency information. As shown in Fig. 4, we observe that: 1) Compared with "FedFAU_V1" and "FedFAU_V2", FedFAU achieves the best performance across all four clustering metrics; 2) Compared with "FedFAU_V1", "FedFAU_V2" demonstrates overall superior performance across the four clustering metrics. These findings can be attributed to the following points. Firstly, block partition captures fine-grained local clustering information; Secondly, frequency information characterizes unlearning-induced changes, enabling selective updates of affected clustering knowledge.

5 Conclusion

In this work, we investigate unlearning in federated graph clustering and propose a novel frequency-aware unlearning method to identify and update clustering structures affected by unlearning. In our method, the blockwise frequency information captures local clustering changes, and the local prototypes are obtained by K-means. Following this, the blockwise frequency information and the local prototypes are uploaded to the global server to identify and update clustering knowledge affected by unlearning. Accordingly, the server leverages local prototypes and frequency information to identify clustering knowledge affected by unlearning and selectively update it. Extensive experiments on five benchmark datasets demonstrate the superiority of FedFAU over existing methods. In future work, we plan to improve the generalization ability of FedFAU and extend it to more challenging unlabeled federated graph unlearning tasks, such as incomplete attributes or imbalanced data distributions.

AI use statement

This manuscript was prepared without the use of generative AI tools.

REFERENCES

Yuming Ai, Xunkai Li, Jiaqi Chao, Bowen Fan, Zhengyu Wu, Yinlin Zhu, Rong-Hua Li, and Guoren Wang. Page: A unified approach for federated graph unlearning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 19589–19597, 2026.

Deyu Bo, Xiao Wang, Chuan Shi, Meiqi Zhu, Emiao Lu, and Peng Cui. Structural deep clustering network. In *Proceedings of the web conference 2020*, pp. 1400–1410, 2020.

Deyu Bo, Xiao Wang, Chuan Shi, and Huawei Shen. Beyond low-frequency information in graph convolutional networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 3950–3957, 2021.

Jinyu Cai, Yunhe Zhang, Jicong Fan, and See-Kiong Ng. Lg-fgad: An effective federated graph anomaly detection framework. In *IJCAI*, pp. 3760–3769, 2024a.

Jinyu Cai, Yunhe Zhang, Zhoumin Lu, Wenzhong Guo, and See-Kiong Ng. Towards effective federated graph anomaly detection via self-boosted knowledge distillation. In *Proceedings of the 32nd ACM international conference on multimedia*, pp. 5537–5546, 2024b.

Jiaqi Chao, Xunkai Li, Yuming Ai, and Meixia Qu. Fedasu: Attention-guided sensitivity unlearning framework for multi-scenario federated graph unlearning. *Expert Systems with Applications*, pp. 132218, 2026.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. Graph neural networks for social recommendation. In *The world wide web conference*, pp. 417–426, 2019.

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.

Lei Gong, Sihang Zhou, Wenxuan Tu, and Xinwang Liu. Attributed graph clustering with dual redundancy reduction. In *IJCAI*, pp. 3015–3021, 2022.

Haoyu Han, Xiaorui Liu, Li Ma, MohamadAli Torkamani, Hui Liu, Jiliang Tang, and Makoto Yamada. Structural fairness-aware active learning for graph neural networks. In *International Conference on Learning Representations*, volume 2024, pp. 51287–51303, 2024.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pp. 639–648, 2020.

Bo Jiang, Ziyan Zhang, Sheng Ge, Beibei Wang, Xiao Wang, and Jin Tang. Learning graph attentions via replicator dynamics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):7720–7727, 2024.

De Li, Haodong Qian, Qiyu Li, Zhou Tan, Zemin Gan, Jinyan Wang, and Xianxian Li. Fedrgl: Robust federated graph learning for label noise. *arXiv preprint arXiv:2411.18905*, 2024a.

Xunkai Li, Zhengyu Wu, Wentao Zhang, Yinlin Zhu, Rong-Hua Li, and Guoren Wang. Fedgta: Topology-aware averaging for federated graph learning. *arXiv preprint arXiv:2401.11755*, 2024b.

Xunkai Li, Yulin Zhao, Zhengyu Wu, Wentao Zhang, Rong-Hua Li, and Guoren Wang. Towards effective and general graph unlearning via mutual evolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 13682–13690, 2024c.

Xunkai Li, Yinlin Zhu, Boyang Pang, Guochen Yan, Yeyu Yan, Zening Li, Zhengyu Wu, Wentao Zhang, Rong-Hua Li, and Guoren Wang. Openfgl: A comprehensive benchmark for federated graph learning. *arXiv preprint arXiv:2408.16288*, 2024d.

Zhiping Lin, Zhao Kang, Lizong Zhang, and Ling Tian. Multi-view attributed graph clustering. *IEEE Transactions on knowledge and data engineering*, 35(2):1872–1880, 2021.

Bingchen Liu and Yuanyuan Fang. Federated knowledge graph unlearning via diffusion model. *arXiv preprint arXiv:2403.08554*, 2024.

Fan Liu and Hao Liu. Subgraph federated unlearning. In *Proceedings of the ACM on Web Conference 2025*, pp. 1205–1215, 2025.

Gaoyang Liu, Xiaoqiang Ma, Yang Yang, Chen Wang, and Jiangchuan Liu. Federaser: Enabling efficient client-level data removal from federated learning models. In *2021 IEEE/ACM 29th international symposium on quality of service (IWQOS)*, pp. 1–10, 2021.

Jingxin Liu, Renda Han, Wenxuan Tu, Haotian Wang, Junlong Wu, and Jieren Cheng. Federated node-level clustering network with cross-subgraph link mending. In *ICML*, 2025.

Yue Liu, Wenxuan Tu, Sihang Zhou, Xinwang Liu, Linxuan Song, Xihong Yang, and En Zhu. Deep graph clustering via dual correlation reduction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 7603–7611, 2022.

Yue Liu, Xihong Yang, Sihang Zhou, Xinwang Liu, Siwei Wang, Ke Liang, Wenxuan Tu, and Liang Li. Simple contrastive graph clustering. *IEEE Transactions on Neural Networks and Learning Systems*, 35(10):13789–13800, 2023a.

Yue Liu, Xihong Yang, Sihang Zhou, Xinwang Liu, Zhen Wang, Ke Liang, Wenxuan Tu, Liang Li, Jingcan Duan, and Cancan Chen. Hard sample aware network for contrastive deep graph clustering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 8914–8922, 2023b.

Lingyuan Meng, Ke Liang, Hao Yu, Yue Liu, Sihang Zhou, Meng Liu, and Xinwang Liu. Fedean: Entity-aware adversarial negative sampling for federated knowledge graph reasoning. *IEEE Transactions on Knowledge and Data Engineering*, 36(12):8206–8219, 2024.

Yimeng Min, Frederik Wenkel, and Guy Wolf. Scattering gcn: Overcoming oversmoothness in graph convolutional networks. *Advances in neural information processing systems*, 33:14498–14508, 2020.

Shirui Pan, Ruiqi Hu, Guodong Long, Jing Jiang, Lina Yao, and Chengqi Zhang. Adversarially regularized graph autoencoder for graph embedding. *arXiv preprint arXiv:1802.04407*, 2018.

Oleg Platonov, Denis Kuznedelev, Michael Diskin, Artem Babenko, and Liudmila Prokhorenkova. A critical look at the evaluation of gnns under heterophily: Are we really making progress? *arXiv preprint arXiv:2302.11640*, 2023.

Nicolò Romandini, Alessio Mora, Carlo Mazzocca, Rebecca Montanari, and Paolo Bellavista. Federated unlearning: A survey on methods, design guidelines, and evaluation metrics. *IEEE Transactions on Neural Networks and Learning Systems*, 36(7):11697–11717, 2024.

Yash Sinha, Murari Mandal, and Mohan Kankanhalli. Distill to delete: Unlearning in graph networks with knowledge distillation. *IEEE Transactions on Neural Networks and Learning Systems*, 2025.

Zixing Song. Graph-based label-efficient learning: When graph-structured data meets limited labels. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 39831–39831, 2026.

Zihan Tan, Guancheng Wan, Wenke Huang, and Mang Ye. Fedssp: Federated graph learning with spectral knowledge and personalized preference. *Advances in Neural Information Processing Systems*, 37:34561–34581, 2024.

Wenxuan Tu, Sihang Zhou, Xinwang Liu, Xifeng Guo, Zhiping Cai, En Zhu, and Jieren Cheng. Deep fusion clustering network. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 9978–9987, 2021.

Wenxuan Tu, Renxiang Guan, Sihang Zhou, Chuan Ma, Xin Peng, Zhiping Cai, Zhe Liu, Jieren Cheng, and Xinwang Liu. Attribute-missing graph clustering network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 15392–15401, 2024.

Chun Wang, Shirui Pan, Ruiqi Hu, Guodong Long, Jing Jiang, and Chengqi Zhang. Attributed graph clustering: A deep attentional embedding approach. *arXiv preprint arXiv:1906.06532*, 2019.

Haozhao Wang, Song Guo, Bin Tang, Ruixuan Li, Yutong Yang, Zhihao Qu, and Yi Wang. Heterogeneity-aware gradient coding for tolerating and leveraging stragglers. *IEEE Transactions on Computers*, 71(4):779–794, 2021.

Haozhao Wang, Yichen Li, Wenchao Xu, Ruixuan Li, Yufeng Zhan, and Zhigang Zeng. Dafkd: Domain-aware federated knowledge distillation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20412–20421, 2023.

Haozhao Wang, Yabo Jia, Meng Zhang, Qinghao Hu, Hao Ren, Peng Sun, Yonggang Wen, and Tianwei Zhang. Feddse: Distribution-aware sub-model extraction for federated learning over resource-constrained devices. In *Proceedings of the ACM web conference 2024*, pp. 2902–2913, 2024.

Xiaobao Wang, Renda Han, Ronghao Fu, and Di Jin. Federated graph-level clustering network with dual knowledge separation. In *The Fourteenth International Conference on Learning Representations*, 2026.

Zichen Wen, Tianyi Wu, Yazhou Ren, Yawen Ling, Chenhang Cui, Xiaorong Pu, and Lifang He. Dual-optimized adaptive graph reconstruction for multi-view graph clustering. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 1819–1828, 2024.

Jiancan Wu, Yi Yang, Yuchun Qian, Yongduo Sui, Xiang Wang, and Xiangnan He. Gif: A general graph unlearning strategy via influence function. In *Proceedings of the ACM web conference 2023*, pp. 651–661, 2023.

Leijie Wu, Song Guo, Junxiao Wang, Zicong Hong, Jie Zhang, and Yaohong Ding. Federated unlearning: Guarantee the right of clients to forget. *IEEE Network*, 36(5):129–135, 2022.

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.

Xihong Yang, Yue Liu, Sihang Zhou, Siwei Wang, Wenxuan Tu, Qun Zheng, Xinwang Liu, Liming Fang, and En Zhu. Cluster-guided contrastive graph clustering network. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 10834–10842, 2023.

Xiangrong Zhu, Guangyao Li, and Wei Hu. Heterogeneous federated knowledge graph embedding learning and unlearning. In *Proceedings of the ACM web conference 2023*, pp. 2444–2454, 2023.

Yinlin Zhu, Xunkai Li, Zhengyu Wu, Di Wu, Miao Hu, and Rong-Hua Li. Fedtad: Topology-aware data-free knowledge distillation for subgraph federated learning. *arXiv preprint arXiv:2404.14061*, 2024.

A Detail of Method

A.1 Notations

In this section, we provide all notations of the proposed FedFAU in Table 3

B Experimental Settings

B.1 Benchmark Datasets

As shown in Table 4, we present the specific information for five benchmark datasets. Detailed descriptions of these datasets are provided below:

Table 3: Basic notations for the proposed FedFAU.

Notations	Meaning
$\bar{\mathbf{X}} \in \mathbb{R}^{\bar{N} \times d}$	The attribute matrix of each subgraph
$\hat{\mathbf{A}} \in \mathbb{R}^{\bar{N} \times \bar{N}}$	Original adjacency matrix of each subgraph
$\tilde{\mathbf{D}} \in \mathbb{R}^{\bar{N} \times \bar{N}}$	Original degree matrix of each subgraph
$\tilde{\mathbf{A}} \in \mathbb{R}^{\bar{N} \times \bar{N}}$	Normalized adjacency matrix of each subgraph
$\bar{\mathbf{S}} \in \mathbb{R}^{B \times \bar{K}}$	The node count matrix of each subgraph
$\bar{\mathbf{A}} \in \mathbb{R}^{\bar{N} \times \bar{N}}$	Blockwise adjacency matrix of each subgraph
$\bar{\mathbf{D}} \in \mathbb{R}^{\bar{N} \times \bar{N}}$	Blockwise degree matrix of each subgraph
$\bar{\mathbf{A}}_b \in \mathbb{R}^{n_b \times n_b}$	Adjacency matrix of the b-th block
$\bar{\mathbf{D}}_b \in \mathbb{R}^{n_b \times n_b}$	Degree matrix of the b-th block
$\bar{\mathbf{Z}}^{(l)} \in \mathbb{R}^{\bar{N} \times d'}$	Node representations at the l-th encoder layer
$\bar{\mathbf{Z}} \in \mathbb{R}^{\bar{N} \times d''}$	Final node representations of each subgraph
$\bar{\mathbf{C}} \in \mathbb{R}^{\bar{K} \times d''}$	Local cluster prototypes
$\bar{\mathbf{Q}} \in \mathbb{R}^{\bar{N} \times \bar{K}}$	Local soft assignment matrix
$\bar{\mathbf{C}}_b \in \mathbb{R}^{\bar{K} \times d''}$	Local prototypes of the b-th block
$\bar{\mathbf{Q}}_b \in \mathbb{R}^{n_b \times \bar{K}}$	Soft assignment matrix of the b-th block
$\mathbf{L}_b \in \mathbb{R}^{n_b \times n_b}$	Graph Laplacian matrix of the b-th block
$\mathbf{U}_b \in \mathbb{R}^{n_b \times n_b}$	Eigenvector matrix of $\mathbf{L}_b$
$\mathbf{\Lambda}_b \in \mathbb{R}^{n_b \times n_b}$	Eigenvalue matrix of $\mathbf{L}_b$
$\hat{\mathbf{Q}}_b \in \mathbb{R}^{n_b \times \bar{K}}$	Graph Fourier coefficient matrix of the b-th block
$\mathbf{R}^L \in \mathbb{R}^{B \times \bar{K}}$	Low-frequency information matrix
$\mathbf{R}^H \in \mathbb{R}^{B \times \bar{K}}$	High-frequency information matrix
$\bar{\mathbf{P}} \in \mathbb{R}^{\bar{N} \times \bar{K}}$	Target distribution by sharpening $\bar{\mathbf{Q}}$

Citation Graphs. CiteSeer, PubMed, and Cora are widely used citation graph datasets, where nodes represent papers and edges denote citation relationships. Node features are derived from textual information, while labels correspond to research categories.

Co-purchase Graphs. Am.Co and Am.Ph are two subsets of the Amazon co-purchase datasets, where nodes represent products and edges indicate co-purchase relationships. Node attributes are constructed from product-related text, and labels denote product categories.

B.2 Algorithm

The main text has presented the overall framework and key components of FedFAU. To further clarify its implementation details, we provide the complete algorithmic procedure in Algorithm 1.

B.3 Baselines

D2DGN (Sinha et al. (2025)) distills the trained model against two separators: a frozen copy of itself supplies the knowledge to keep in the retained set, while a randomly initialized model provides neutral targets to suppress the knowledge of the forget set. No sharding or deletion layer is introduced, so inference cost remains unchanged.

FedDM (Liu & Fang (2024)) introduces diffusion-based unlearning for federated KG embeddings. It learns to generate noisy representations of forgotten knowledge and replaces the original embeddings with these generated representations, diluting the forgotten features while maintaining the performance on remaining knowledge.

FedLU (Zhu et al. (2023)) uses mutual distillation between local and global embeddings to absorb client heterogeneity during training. Unlearning then erases the target triples locally via retroactive interference and passive decay, and reuses the same distillation channel to propagate the effect to the global embedding.

GIF (Wu et al. (2023)) adds a loss term over the k-hop neighborhoods affected by a deletion, so the estimated parameter shift also captures GNN propagation. The resulting closed-form update covers node-, edge-, and feature-level requests without retraining.

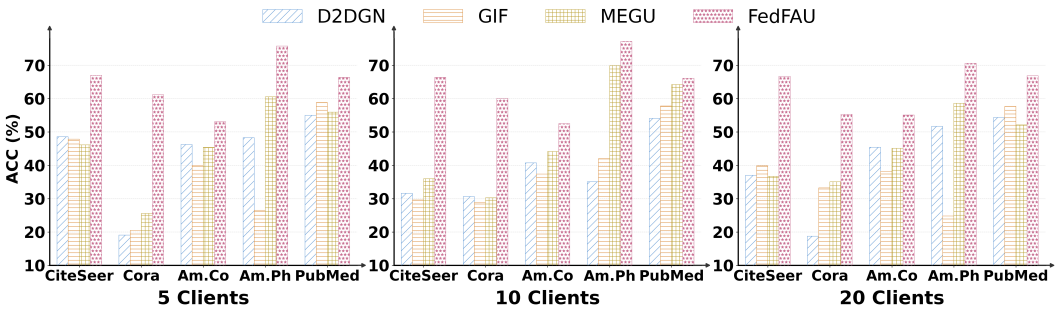

Figure 5: Performance comparison of FedFAU with three centralized unlearning methods on five graph datasets.

Table 4: Dataset summary.

Datasets	Nodes	Edges	Dimensions	Clusters	Type
CiteSeer	3,327	9,104	3,703	6	Citation Graph
Cora	2,708	10,556	1,433	7	Citation Graph
PubMed	19,717	88,648	500	3	Citation Graph
Am.Co	13,752	491,722	767	10	Co-purchase Graph
Am.Ph	7,650	238,162	745	8	Co-purchase Graph

MEGU (Li et al. (2024c)) trains the predictive and unlearning modules in one loop so that forgetting and predictive accuracy improve together. Instead of a fixed k-hop neighborhood, it inverts the deleted entities' features and uses the influence distance between forward and inverse propagation to select the highly influenced nodes adaptively.

PAGE (Ai et al. (2026)) handles both meta-level and client-level unlearning in federated graph learning. It combines prototype matching, adversarial graph generation, and negative knowledge distillation to remove target knowledge and reduce its remaining influence on other clients.

B.4 Comparison with Supervised Unlearning Methods.

In the main text, we compare the proposed FedFAU with FedLU, FedDM, and PAGE, demonstrating the superiority of our method. Here, we compare it with three supervised centralized unlearning methods: D2DGN, MEGU, GIF, aiming to further confirm the competitiveness of FedFAU in clustering work. The experimental setup remains consistent with that in the main text, where each compared method is trained using only 2% labeled data. As shown in Fig. 5, the results show that FedFAU consistently delivers higher ACC across various datasets and client partition settings, demonstrating its effectiveness for node unlearning.

Algorithm 1 The learning procedure of FedFAU

Input: Complete graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$. For each subgraph: attribute matrix $\bar{\mathbf{X}}$; adjacency matrix $\tilde{\mathbf{A}}$; degree matrix $\tilde{\mathbf{D}}$; server epochs E_{server}; client epochs E_{client}.
Output: Clustering results $\mathbf{Y}$ at clients.
Initialize the $\bar{\theta}$ of all clients.
for $t = 1$ to E_{server} **do**
  `// Blockwise Frequency Extraction`
  for $l = 1$ to E_{client} **do**
    Obtain $\bar{\mathbf{A}}_b$ and $\bar{\mathbf{D}}_b$ from $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{D}}$ by Eq(1).
    Obtain $\tilde{\mathbf{Z}}$ from $\tilde{\mathbf{A}}$ and $\bar{\mathbf{X}}$ by GNN.
    Obtain $\{\bar{\mathbf{C}}_b\}_{b=1}^{B}$, $\{\bar{\mathbf{Q}}_b\}_{b=1}^{B}$ and $\bar{\mathbf{S}}$ by K-means.
    Calculate $\mathbf{L}_b$ from $\bar{\mathbf{A}}_b$ and $\bar{\mathbf{D}}_b$.
    Obtain $\mathbf{U}_b$ and $\mathbf{\Lambda}_b$ from $\mathbf{L}_b$ by Eq(2).
    Calculate $\hat{\mathbf{Q}}_b$ from $\mathbf{U}_b$ and $\bar{\mathbf{Q}}_b$.
    Generate $\mathbf{R}^L$ and $\mathbf{R}^H$ from $\hat{\mathbf{Q}}_b$ and $\mathbf{\Lambda}_b$ by Eq(3).
    Train local model $\bar{\theta}$ by Eqs.(4-6)
    Upload $\bar{\theta}$, $\bar{\mathbf{S}}$,$\{\bar{\mathbf{C}}_b\}_{b=1}^{B}$, $\mathbf{R}^L$ and $\mathbf{R}^H$
  end for
  `// Frequency-Guided Graph Unlearning`
  Obtain $\hat{\mathbf{C}}$ and θ_g by FedAVG.
  calculate $\eta_{b,k}$ from $r_{b,k}^L$ and $r_{b,k}^H$.
  Calculate $\zeta_{b,k}$ from $\eta_{b,k}^{\text{old}}$ and $\eta_{b,k}^{\text{new}}$ by Eq(7).
  if $\zeta_{b,k}$ = 1 **then**
    update $\hat{\mathbf{C}}$ and θ_g by Eq(8) and Eq(9).
  end if
  Backpropagate $\hat{\mathbf{C}}$ and θ_g to each client.
end for
Obtain clustering results $\mathbf{Y}$ at clients by K-means.
return Y