GRAPH FEDERATED UNLEARNING FOR PRIVACY PRESERVATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph federated learning (GFL) facilitates decentralized training on distributed graph data while keeping sensitive user information local, aligning with policies such as GDPR and CCPA that grant users the right to freely join or withdraw from learning systems. However, even decentralized, user information can persist after quitting, potentially propagating to central servers and then redistributing to malicious clients. This privacy leakage during user withdrawal, despite its importance, has received seldom attention in GFL. To fill the gap, we explore the potential of *machine unlearning* (MU) to thoroughly remove user information. However, classical MU methods are known to degrade overall performance, a problem that is exacerbated in GFL due to local message passing and global model collaboration. To this end, we make two adjustments to mitigate this challenge for GFL. First, we ensure unlearning updates that minimally affect overall performance, steering them in directions orthogonal to the gradients from learning other data. Second, we introduce virtual clients, maintained by the central server, to preserve graph topology and global embeddings without recovering information of removed entities. We conduct comprehensive experiments under a representative user-withdrawal scenario and propose a novel membership inference framework to rigorously evaluate and validate the reliability of our privacy preservation. The experimental results demonstrate the effectiveness of our approach, which also surpasses the performance of seven state-of-the-art baseline methods. The code is available at: `https://anonymous.4open.science/r/FedGCV_FGMU/`.

1 INTRODUCTION

Graph-structured data are pervasive in modern AI applications, and *graph neural networks* (GNNs) Xu et al. (2019); Wu et al. (2020); Luan et al. (2022); Corso et al. (2024); Yu et al. (2026) have been extensively studied as a powerful architecture for analyzing such data, leveraging message-passing mechanisms to capture both node attributes and inherent topological structures. Further with recent advances in spectral graph theory Bo et al. (2023) and attention mechanisms Ma et al. (2025), GNNs are now capable of supporting large-scale, real-world applications ranging from high-frequency financial trading Li et al. (2024) to industrial recommender systems Ko et al. (2022). However, processing such large volumes of data with message propagation, especially in data-sensitive domains, makes GNNs particularly susceptible to privacy risks, thereby hindering their broader deployments Liu et al. (2021).

In response to increasingly stringent data-sharing regulations, *graph federated learning* (GFL) Zhou et al. (2024); Yu et al. (2025a) has emerged as a promising solution. GFL enables decentralized GNN training, where clients locally update models and collaboratively contribute to a central server without sharing raw data. It aligns better with privacy regulations Voigt & Von dem Bussche (2016). However, such improvements do not fully eliminate privacy risks, namely gradient information and model parameters remain vulnerable to membership inference Liu et al. (2021) and inversion attacks He et al. (2019). In this paper, we focus on an important yet underexplored implication of this vulnerability: *GFL cannot fully satisfy the "right to be forgotten" (RTBF)* Voigt & Von dem Bussche (2016).

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.

RTBF grants individuals the right to request the deletion of their personal information from databases and AI systems. It is a cornerstone of modern data protection laws, including GDPR Voigt & Von dem Bussche (2016), and has been adopted by regulations such as CCPA Bonta (2022), LGPD de Oliveira et al. (2025), and DPDPA Dave & Agrawal (2025). In GFL, RTBF requires that users can freely join or withdraw, with their associated information correspondingly embedded into or removed from GNN model parameters. However, as we will show in Section 4.2 simply removing clients is insufficient, as membership inference Liu et al. (2021) reveals that associated information can still be inferred from other clients due to parameter embedding and propagation.

To mitigate such privacy leakage during user withdrawal, we explore *machine unlearning* (MU) Cao & Yang (2015) for RTBF. Classical MU methods typically assume that a small portion of training data needs to be removed and aim to eliminate its influence, including partition-based unlearning Bourtoule et al. (2021) and optimization-based approximations Chen et al. (2021); Tarun et al. (2023); Wang et al. (2025b). However, they incur high computational cost and may degrade overall performance. More recent *gradient-ascent* (GA)-based methods Fan et al. (2023); Huang et al. (2024); Liu et al. (2025); Wang et al. (2025a); Li et al. (2025) directly suppress predictions on target data and are more practical. However, they still often harm overall model performance. This issue is more severe in GFL, where unlearning effects can propagate and accumulate across local nodes and spread through federated aggregation, affecting non-target clients. Therefore, existing GA-based methods require further improvement for RTBF in the context of GFL.

To this end, we propose a *graph **Fed**erated unlearning method with **G**radient **C**orrection and **V**irtual client (FedGCV)*, an enhanced GA-based MU framework tailored for graph federated unlearning (GFU). To mitigate the negative impact on overall performance, FedGCV introduces two key improvements over previous MU methods. First, to counteract the erroneous propagation of unlearning effects to normal clients that are not intended to be removed, we introduce **gradient correction** to explicitly preserve overall GNN performance. Specifically, we constrain the unlearning gradients to be orthogonal to that used for learning other clients, which can be coordinated by the central servers and justified as performance-preserving under a first-order approximation. Second, we further preserve topological structures by introducing **virtual clients**, also organized by the central server, to replace the unlearned clients and synthesize graphs to maintain both node attributes and inherent topological structures. The virtual clients preserve graph topologies and global embeddings while potentially retaining information from removed entities, which can be employed to preserve the performance of the global model. Therefore, GFU combines gradient correction and virtual clients, aiming to enhance performance preservation while maintaining reliable RTBF.

We further propose a membership-inference-based evaluation framework in section 4.1 to assess the degree of privacy preservation. Moreover, this is the first work to employ MIA evaluation metric in the GFU scenario. Through extensive experiments across a representative RTBF setup for GFU, we demonstrate the reliability of our method in preserving overall model performance meanwhile mitigating privacy leakage.

2 PRELIMINARIES

In this section, we briefly review the relevant background. First, we introduce GNNs for graph-structured data, in Section 2.1. Then, we discuss the use of federated learning with GNNs, leading to GFL, in Section 2.2. Finally, we highlight the drawbacks of GFL, particularly its limited flexibility in meeting advanced data-sharing regulations, in Section 2.3.

2.1 GRAPH NEURAL NETWORKS

GNNs extend conventional deep neural networks to graph-structured data via message passing Bai et al. (2022); Yu et al. (2023). Generally, node representations are iteratively updated by aggregating information from their neighborhoods. After k iterations, a node captures structural information within its k-hop neighborhood. Formally, the representation of node v_i at the k-th layer is

$$h_{v_i}^{(k)} = \mathrm{CO}^{(k)}\left(h_{v_i}^{(k-1)}, \mathrm{AG}^{(k)}(\{h_{v_j}^{(k-1)} : v_j \in \mathcal{N}(v_i)\})\right). \tag{1}$$

Here, $\mathcal{N}(v_i)$ denotes the neighbors of v_i, $\mathrm{AG}^{(k)}$ is a permutation-invariant aggregation operator (e.g., sum, mean, max, or attention-weighted sum Vaswani et al. (2017)), and $\mathrm{CO}^{(k)}$ is a learnable mapping function such as a multilayer perceptron or a gated unit Cho et al. (2014).

Such a message-passing mechanism has achieved great success in modeling graph-structured data and is highly adaptable to large-scale settings with complex user relationships, ranging from high-frequency financial trading Li et al. (2024) to industrial recommender systems Ko et al. (2022).

2.2 Graph Federated Learning

Despite their effectiveness, GNNs typically require centralized data collection for message passing which raises data-sharing and privacy concerns Voigt & Von dem Bussche (2016). This motivates their integration with federated learning Li et al. (2020), leading to graph federated learning (GFL) Zhou et al. (2025); Yu (2025); Yu et al. (2025b), which enables collaborative training a global model across multiple clients without sharing raw data, thereby preserving data privacy.

Formally, we consider a federated setup with a central server S and M clients, where each client $i \in \{1, 2, \cdots, M\}$ holds a local graph $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i)$ with a feature matrix $\mathbf{X}_i$ and a label matrix $\mathbf{Y}_i$, where $\mathcal{V}_i$ is the node set and $\mathcal{E}_i$ is the edge set. The goal is to learn model parameters $\mathbf{w}$ by minimizing the empirical risk over all clients:

$$\begin{aligned} \hat{\mathbf{w}}^* &= \arg\min_{\mathbf{w}} \hat{\mathcal{L}}(\mathbf{w}; \{\mathcal{G}_i, \mathbf{X}_i, \mathbf{Y}_i\}_{i=1}^M) \\ &= \sum_{i=1}^{M} p_i \hat{\mathcal{L}}_i(\mathbf{w}; \mathcal{G}_i, \mathbf{X}_i, \mathbf{Y}_i), \end{aligned} \tag{2}$$

where p_i is the aggregation weight, typically defined to reflect the relative importance or data size of client i, and $\hat{\mathcal{L}}_i(\mathbf{w}; \mathcal{G}_i, \mathbf{X}_i, \mathbf{Y}_i)$ is the local empirical risk on client i, typically defined as the cross-entropy loss.

In practice, this objective is optimized via FedAvg McMahan et al. (2017). At each round t, the server broadcasts $\mathbf{w}^{(t)}$, clients perform local updates to obtain $\mathbf{w}_i^{(t+1)}$, and the server aggregates them as $\mathbf{w}^{(t+1)} = \sum_{i=1}^{M} p_i \mathbf{w}_i^{(t+1)}$ over T communication rounds.

2.3 Right To Be Forgotten

GFL is better aligned with data-sharing regulations than its centralized counterparts, as data are processed locally on clients and never directly shared with the central server. However, with increasing attention to privacy, new requirements must be fulfilled, such as the "*right to be forgotten*" (RTBF) Voigt & Von dem Bussche (2016); Bonta (2022). More detailed introduction is explained in Appendix A.3.1

2.4 Machine Unlearning

Machine unlearning (MU) Cao & Yang (2015) has been widely studied as a practical approach to achieve RTBF. Early work is rooted in differential privacy (DP) Abadi et al. (2016), aiming to ensure that the unlearned model behaves similarly to one trained without the target data. For instance, Golatkar et al. (2020) propose adding Gaussian noise shaped by the Hessian:

$$\hat{\mathbf{w}}^{\mathrm{u}} = \hat{\mathbf{w}}^* + (\lambda_0 \sigma^2)^{1/4} \mathbf{H}_{\mathrm{r}}^{-1/4} n, \text{ where } n \sim \mathcal{N}(0, 1), \tag{3}$$

with λ_0 a trade-off hyper-parameter, σ a prior variance, and H_{r} the Hessian matrix with respect to $\hat{\mathcal{L}}(\mathbf{w}; \{\mathcal{G}_i, \mathbf{X}_i, \mathbf{Y}_i\}_{i=1}^M \backslash \{\mathcal{G}_j, \mathbf{X}_j, \mathbf{Y}_j\})$. Overall, the fractional Hessian inverse $H_{\mathrm{r}}^{-1/4}$ shapes the noise to be aligned with the local curvature, mitigating performance degradation while achieving the desired level of privacy. Guo et al. (2020) certifies that natural gradient updates, following

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + \mathbf{H}_{\mathrm{u}}^{-1} \nabla_{\mathbf{w}} \hat{\mathcal{L}}(\mathbf{w}; \mathcal{G}_j, \mathbf{X}_j, \mathbf{Y}_j). \tag{4}$$

at each unlearning step t, where $\mathbf{H}_{\mathrm{u}}$ is the Hessian matrix for $\hat{\mathcal{L}}(\mathbf{w}; \mathcal{G}_j, \mathbf{X}_j, \mathbf{Y}_j)$. Generally, DP-based unlearning methods enjoy relatively strong theoretical guarantees. However, on the other side,

they typically rely on linear assumptions, require tedious hyper-parameter tuning, and often cause substantial degradation in overall performance.

More recent approaches adopt a practical formulation by directly increasing the risk on the target data, leading to gradient ascent (GA)-based methods Fan et al. (2023); Wang et al. (2025a). These methods further preserve the model performance on other data that are not intended to be unlearned and incorporate a retention objective:

$$\arg\min_{\mathbf{w}} -\hat{\mathcal{L}}(\mathbf{w}; \mathcal{G}_j, \mathbf{X}_j, \mathbf{Y}_j) + \lambda \hat{\mathcal{L}}(\mathbf{w}; \{\mathcal{G}_i, \mathbf{X}_i, \mathbf{Y}_i\}_{i=1}^{M} \backslash \{\mathcal{G}_j, \mathbf{X}_j, \mathbf{Y}_j\}), \tag{5}$$

where λ is a trade-off hyper-parameter. Although less theoretically grounded, GA-based methods have been shown to achieve more efficient and reliable MU effects than DP-based methods in practice. However, since $\hat{\mathcal{L}}$ is unbounded above, maximizing $\hat{\mathcal{L}}(\mathbf{w}; \mathcal{G}_j, \mathbf{X}_j, \mathbf{Y}_j)$, or equivalently minimizing $-\hat{\mathcal{L}}(\mathbf{w}; \mathcal{G}_j, \mathbf{X}_j, \mathbf{Y}_j)$, can lead to gradient explosion Wang et al. (2025a), thereby degrading performance.

This issue is further amplified in GFL, where information propagates across clients through the exchanged model parameters Zhang et al. (2021). As a result, unlearning updates may affect a broader set of data and distort graph structures, potentially harming the overall quality of the global model, which makes effective unlearning in GFL particularly challenging. Therefore, we need further improve upon existing GA-based unlearning methods to make it more suitable for GFL scenarios

3 Our Proposed Method

In this section, we propose ***Fed**erated unlearning method with **G**radient **C**orrection and **V**irtual client* (FedGCV), an enhanced GA-based MU method specifically designed for GFL. FedGCV makes two primary contributions. First, we introduce a **gradient correction** strategy that explicitly addresses the unlearning-preservation trade-off. Second, we propose a **virtual client** strategy that aims to preserve the original topological structure. Together, these components yield a reliable RTBF framework that better preserves overall model performance than conventional GA-based methods such as Eq. equation 5 for GFL.

3.1 Gradient Correction

Conventional GA-based MU formulates unlearning as a bi-objective problem, which maximizes the risk on the target data, i.e., $\max_{\mathbf{w}} \hat{\mathcal{L}}(\mathbf{w}; \mathcal{G}_j, \mathbf{X}_j, \mathbf{Y}_j)$, and minimizes the risk on the remaining data, i.e., $\min_{\mathbf{w}} \hat{\mathcal{L}}(\mathbf{w}; \{\mathcal{G}_i, \mathbf{X}_i, \mathbf{Y}_i\}_{i=1}^{M} \setminus \{\mathcal{G}_j, \mathbf{X}_j, \mathbf{Y}_j\})$. However, previous work has shown that these two objectives are inherently conflicting Wang et al. (2025b), and tuning λ only amplifies this trade-off, leading to either insufficient unlearning or degraded model performance. Therefore, we seek an alternative formulation beyond Eq. equation 5, that can effectively mitigate this trade-off.

Our key motivation is to ensure that each unlearning update has minimal negative impact on the performance of other normal data, leading to a constrained optimization problem

$$\begin{aligned} &\arg\min_{\mathbf{w}} -\hat{\mathcal{L}}(\mathbf{w}; \mathcal{G}_j, \mathbf{X}_j, \mathbf{Y}_j) \\ \text{s.t. } &\hat{\mathcal{L}}(\mathbf{w}; \{\mathcal{G}_i, \mathbf{X}_i, \mathbf{Y}_i\}_{i=1}^{M} \backslash \{\mathcal{G}_j, \mathbf{X}_j, \mathbf{Y}_j\}) \\ &\leq \hat{\mathcal{L}}(\hat{\mathbf{w}}^*; \{\mathcal{G}_i, \mathbf{X}_i, \mathbf{Y}_i\}_{i=1}^{M} \backslash \{\mathcal{G}_j, \mathbf{X}_j, \mathbf{Y}_j\}). \end{aligned} \tag{6}$$

This constrained objective is a heuristic: target-loss ascent and a retained-loss bound do not prove retraining equivalence. Appendix A.3.2 analyzes a local retained-utility property. Define the target-loss ascent direction $\mathbf{a}_{\mathrm{u}} = \nabla \hat{\mathcal{L}}_{\mathrm{u}}$ and retained gradient $\mathbf{g}_{\mathrm{r}} = \nabla \hat{\mathcal{L}}_{\mathrm{r}}$. A step $+\eta \mathbf{a}_{\mathrm{u}}$ changes retained loss to first order by $\eta \langle \mathbf{g}_{\mathrm{r}}, \mathbf{a}_{\mathrm{u}} \rangle$. We impose

$$\langle \mathbf{g}_{\mathrm{r}}, \widetilde{\mathbf{a}}_{\mathrm{u}} \rangle \leq 0. \tag{7}$$

The closest feasible update in Euclidean norm is

$$\widetilde{\mathbf{a}}_{\mathrm{u}} = \mathbf{a}_{\mathrm{u}} - \frac{\max\{\langle \mathbf{a}_{\mathrm{u}}, \mathbf{g}_{\mathrm{r}} \rangle, 0\}}{\|\mathbf{g}_{\mathrm{r}}\|_2^2} \mathbf{g}_{\mathrm{r}}. \tag{8}$$

Set $\widetilde{\mathbf{a}}_{\mathrm{u}} = \mathbf{a}_{\mathrm{u}}$ when $\mathbf{g}_{\mathrm{r}} = 0$. This projects the update relative to the retained gradient, not model parameters into the kernel of a sensitive gradient.

For a FedAvg round from $\mathbf{w}^{(t)}$, let $\mathbf{d}_i = \mathbf{w}_i^{(t+1)} - \mathbf{w}^{(t)}$ and $\bar{p}_i = p_i / \sum_{k \neq j} p_k$. The retained update is

$$\boldsymbol{\Delta}_{\mathrm{r}} = \sum_{i \neq j} \bar{p}_i \mathbf{d}_i. \tag{9}$$

Let $\boldsymbol{\Delta}_{\mathrm{u}}$ be the actual target-loss ascent update. Approximating $\mathbf{g}_{\mathrm{r}} \propto -\boldsymbol{\Delta}_{\mathrm{r}}$ gives the compatible server rule

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + \boldsymbol{\Delta}_{\mathrm{r}} + \beta \left(\boldsymbol{\Delta}_{\mathrm{u}} - \frac{\min\{\langle \boldsymbol{\Delta}_{\mathrm{u}}, \boldsymbol{\Delta}_{\mathrm{r}} \rangle, 0\}}{\|\boldsymbol{\Delta}_{\mathrm{r}}\|_2^2} \boldsymbol{\Delta}_{\mathrm{r}} \right), \tag{10}$$

where $\beta \geq 0$ and the quotient is zero if $\boldsymbol{\Delta}_{\mathrm{r}} = 0$.

3.2 Virtual Client

After removing a client's subgraph data, both its topology and features are lost, which can notably degrade global performance. Therefore, in the context of FedGCV, gradient correction alone may be insufficient. To restore federated learning performance without revealing the original subgraph data, we introduce *virtual clients*, which synthesize graphs resembling the removed client while potentially retaining information from its data. Specifically, we extract spectral and feature statistics and employ a variational graph autoencoder (VGAE) Kipf & Welling (2016) to generate synthetic subgraphs for subsequent training. This utility-recovery step requires separate evaluation of leakage through the model and derived artifacts.

3.2.1 Extraction of spectral information

To construct virtual clients, we employ a VGAE to capture the latent distribution of the unlearned client. Moreover, to reconstruct the topological structures of the unlearned client, we further employ spectral constraints to guide the generation of edges between nodes across clients.

Let the subgraph on a forgotten client be $\mathcal{G}_j = (\mathcal{V}_j, \mathcal{E}_j)$, $n = |\mathcal{V}_j|$, the adjacency matrix be $\mathbf{A}_j \in \mathbb{R}^{n \times n}$, and the node feature matrix be $\mathbf{X}_j \in \mathbb{R}^{n \times d}$. First, we employ propagation matrices, which are consistent with the message-passing mechanism:

$$\tilde{\mathbf{A}}_j = \mathbf{A}_j + \mathbf{I}, \quad \hat{\mathbf{A}}_j = \tilde{\mathbf{D}}_j^{-\frac{1}{2}} \tilde{\mathbf{A}}_j \tilde{\mathbf{D}}_j^{-\frac{1}{2}}. \tag{11}$$

To capture the structural topologies of forgotten subgraphs, we then extract information from their Laplacian matrices. The graph Laplacian matrix $\mathbf{L}_j$ of subgraph $\mathcal{G}_j$ is defined as $\mathbf{L}_j = \mathbf{D}_j - \mathbf{A}_j$, where $\mathbf{D}_j$ is the degree matrix of $\mathcal{G}_j$ and $\mathbf{A}_j$ is the adjacency matrix of $\mathcal{G}_j$. After performing normalization on the Laplace matrix, we can obtain the normalized Laplacian matrix $\mathbf{L}_j^{\text{norm}}$ as:

$$\mathbf{L}_j^{\text{norm}} = \mathbf{I} - \mathbf{D}_j^{-\frac{1}{2}} \mathbf{A}_j \mathbf{D}_j^{-\frac{1}{2}}. \tag{12}$$

Eq. equation 12 eliminates the influence of the degree of the Laplacian matrix, which further ensures the graph spectrum is bounded and yields more stable numerical results.

Since $\mathbf{L}_j^{\text{norm}}$ is a semi-definite matrix, we can perform an eigenvalue decomposition on $\mathbf{L}_j^{\text{norm}}$ to obtain its eigenvalues and eigenvectors. The equation can be expressed as $\mathbf{L}_j^{\text{norm}} = \mathbf{U}_j \boldsymbol{\Lambda}_j \mathbf{U}_j^{\top}$, and take the first k smallest eigenvalues (low-frequency component) $\boldsymbol{\Lambda}_{jk}$ and their corresponding eigenvectors $\mathbf{U}_{jk}$. The low-frequency subspace characterizes the global connectivity and smooth structure of the subgraph, which is crucial for preserving graph spectral information during generation.

3.2.2 VGAE Modeling

To capture the node feature distributions of the forgotten graph, VGAE learns latent representations of subgraphs in an unsupervised manner so as to generate a similar but privacy-protective virtual subgraph.

Although VGAE is capable of capturing node feature distribution, it may not learn the structural topologies of graphs well. To mind this gap, we exploit the extracted spectral properties in section 3.2.1 and integrate them in the synthesis process of virtual graphs. To preserve the spectral

characteristics of the forgotten graph in the synthetic graph, we project the latent variables onto the subspace spanned by $\mathbf{U}_k$. The projected latent variable $\mathbf{Z}_{\text{proj}}$ can be obtained from the following formula:

$$\mathbf{Z}_{\text{proj}} = \mathbf{U}_k \mathbf{U}_k^\top \mathbf{Z}, \tag{13}$$

where $\mathbf{Z}$ represents the inferred latent distribution of the forgotten graph. $\mathbf{Z}$ is the output of the VGAE encoder, which uses a GCN-based architecture to infer the node feature distribution $q(Z|X, A) = \mathcal{N}(\mu, \sigma^2 I)$. Note that Eq. equation 13 only permits $\mathbf{Z}$ to utilize structures expressible by the original subgraph's spanned subspace. Afterwards, we can obtain the synthetic graphs by performing graph reconstruction based on $\mathbf{Z}_{\text{proj}}$. This reconstruction process is implemented as an inner product following Kipf & Welling (2016). To be specific, this process uses $\mathbf{Z}_{\text{proj}}$ to obtain the edge probability matrix $\mathbf{P}$:

$$\mathbf{P} = \sigma(\mathbf{Z}_{\text{proj}} \mathbf{Z}_{\text{proj}}^\top). \tag{14}$$

Subsequently, we employ a thresholding strategy to generate the synthetic adjacency matrix $\mathbf{A}_{\text{syn}}$ from $\mathbf{P}$, converting continuous probability into a matrix composed of only 0 and 1. The process can be expressed as follows:

$$(\mathbf{A}_{\text{syn}})_{ij} = \mathbb{I}\big[\mathbf{P}_{ij} > \gamma\big], \tag{15}$$

where $\mathbb{I}[\cdot]$ is an indicator function, and $\gamma \in (0, 1)$ is the probability threshold used to determine whether to generate an edge.

Finally, to generate a virtual subgraph, we still need to integrate features with the mean vector $\boldsymbol{\mu}$ and standard deviation vector $\boldsymbol{\sigma}$ of the original subgraph inferred by VGAE. To avoid leaking the original subgraph data $\mathcal{G}$, we also employ a high-privacy statistical sampling strategy to obtain the feature matrix $\mathbf{X}_{\mathbf{syn}}$ reflecting the node feature distribution of the virtual subgraph. Each element $(\mathbf{X}_{\mathbf{syn}})_{ij}$ follows a Gaussian distribution that satisfies the mean and standard deviation of the original subgraph:

$$(\mathbf{X}_{\text{syn}})_{ij} \sim \mathcal{N}\big(\boldsymbol{\mu}, \text{diag}(\boldsymbol{\sigma}^2)\big). \tag{16}$$

Only the node-specific distributions are retained without directly using the original data $\mathcal{G}$, which significantly preserves the privacy of the forgotten graph. The generated $\mathbf{A}_{\text{syn}}$ incorporates the structure of the synthetic subgraph, describing the topological structure of the virtual subgraph, and the generated $\mathbf{X}_{\text{syn}}$ contains the node attributes and features of the synthetic subgraph, describing the node content. Both $\mathbf{A}_{\text{syn}}$ and $\mathbf{X}_{\text{syn}}$ can collectively represent an attributed graph $\mathcal{G}_{\text{virtual}}$ usable in GFL, enabling performance compensation functionality in subsequent parameter updates.

4 Experiments

In this section, we evaluate our proposed methodological framework and compare it against state-of-the-art federated unlearning methods on four widely used graph datasets.

4.1 Experimental Setups

We begin by detailing the experimental setups adopted in this paper, including datasets, evaluation protocols, implementation settings, and baseline methods.

Datasets. Each client possesses a subgraph that is a subset of the original large graph. To be specific, we utilize four real-world datasets that are widely employed in graph learning tasks, including Cora, CiteSeer Yang et al. (2016), PubMed Yang et al. (2016), and Tolokers Platonov et al. (2023). Following prior works Yu (2025), we simulate the real GFU scenario by partitioning the original graph of the mentioned datasets with the METIS algorithm Karypis & Kumar (1998). Detailed information about these datasets is provided in Appendix A.1.1.

Evaluation Protocols. Our evaluation metrics align with the goal of graph federated unlearning, i.e., achieving both effective unlearning and strong model utility. (1) *Membership inference attack* (MIA) Sablayrolles et al. (2019): MIA is the definitive empirical metric for unlearning effectiveness. Nguyen et al. (2024) MIAs evaluate privacy leakage and unlearning effectiveness by assessing whether an adversary can distinguish training samples from non-members based on model outputs. A lower MIA rate indicates better unlearning and privacy protection. Under this paradigm, we adopt MAIT as an evaluation metric, which determines membership by exploiting differences in loss and

Table 1: Overall performance evaluation of graph federated unlearning. **The percentage next to the dataset name indicates the model accuracy achieved by FedAvg without unlearning.**

	Cora (82.42%)		CiteSeer (70.14%)	
Methods	Accuracy ↑	MIA Rate ↓	Accuracy ↑	MIA Rate ↓
retrain	68.67%	**3.70%**	61.83%	**0.00%**
PGA Halimi et al. (2022)	71.91%	12.96%	56.91%	12.70%
SGA-Random	68.09%	14.81%	48.88%	14.29%
SGA-Degree	68.80%	12.22%	51.82%	9.52%
EWC-SGA Wu et al. (2022)	66.89%	9.26%	42.97%	22.22%
Noisy-GD Chourasia & Shah (2023)	72.42%	87.04%	59.62%	88.89%
ULKD	70.79%	18.52%	63.23%	15.87%
ReGEnUnlearn Liu & Liu (2025)	72.04%	9.09%	63.25%	12.39%
FedGCV (Ours)	**79.41%**	5.56%	**69.07%**	7.94%
	PubMed (86.52%)		Tolokers (78.10%)	
Methods	Accuracy ↑	MIA Rate ↓	Accuracy ↑	MIA Rate ↓
retrain	**85.73%**	**0.52%**	**76.58%**	2.89%
PGA Halimi et al. (2022)	76.66%	6.81%	65.65%	11.30%
SGA-Random	75.62%	10.65%	67.77%	5.34%
SGA-Degree	75.70%	9.69%	65.74%	9.46%
EWC-SGA Wu et al. (2022)	77.07%	10.47%	67.55%	7.64%
Noisy-GD Chourasia & Shah (2023)	80.67%	7.07%	66.08%	1.85%
ULKD	82.54%	8.12%	62.57%	10.96%
ReGEnUnlearn Liu & Liu (2025)	77.41%	7.92%	67.47%	1.03%
FedGCV (Ours)	83.35%	3.66%	69.17%	**0.18%**

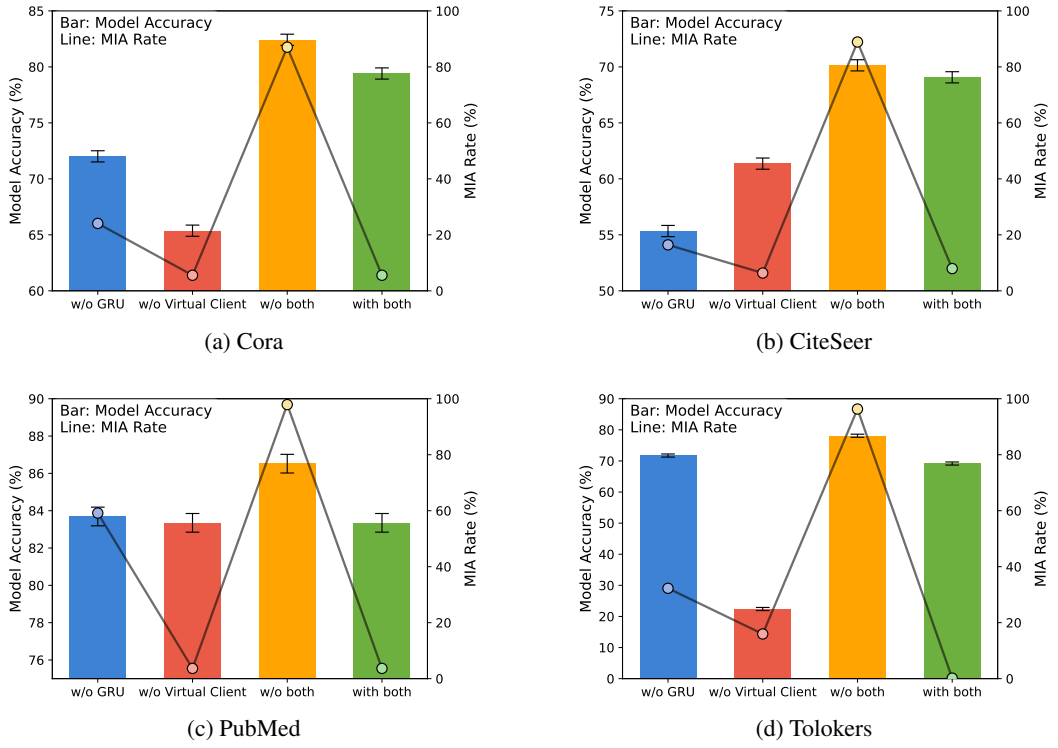
gradient evolution during model training, and quantifies unlearning from the reduction in such distinguishability. Implementation details are provided in Appendix A.1.3. (2) *Model accuracy* (MA): After unlearning, we report the global model accuracy to assess overall utility, where higher MA indicates better performance retention.

Implementation Settings. We set the random seed to $s = 2025$ and train a 2-layer GCN with hidden dimension $d = 64$, dropout rate $p = 0.5$, learning rate $\eta = 10^{-2}$, weight decay $\lambda = 5 \times 10^{-4}$, and batch size $B = 128$ for federated training. We run federated learning for $R = 30$ communication rounds over $K = 10$ clients with full participation ratio $\rho = 1.0$ (i.e., all clients are selected in each round). For interactive unlearning, we optimize the unlearning model for $E_u = 30$ epochs with learning rate $\eta_u = 2 \times 10^{-2}$ and dropout $p_u = 0.3$. We adopt the NPO-based unlearning loss Zhang et al. (2024), an advanced GA-based method, with temperature parameter $\beta = 5.0$ and scale the corrected unlearning update by $s_f = 50$. To reduce catastrophic drift during unlearning, we clip the per-step update norm by $c_{\max} = 10.0$ and project the parameter drift to satisfy $\|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|_2 \leq \tau$, where $\tau = 10.0$. For Graph-MIA-aware training, we additionally enforce a fixed-threshold alignment by pushing the target loss above the fixed-pre threshold by a margin $m = 0.5$ with weight $\lambda_m = 3.0$. For post-unlearning virtual repair, we conduct $R_v = 5$ extra federated rounds. The adjacency reconstruction uses a threshold $\tau_a = 0.7$, and feature perturbation uses noise standard deviation $\sigma_x = 0.1$. We also investigate situations involving an increase in the number of clients and modifications to the GNN architectures, including GAT and GraphSAGE. The experimental results are presented in Appendix A.4.2 and Appendix A.4.3.

Baselines. All the baselines we use in experiments are described in detail in Appendix A.1.2.

4.2 Main Results

table 1 presents the comprehensive results of our proposed method and corresponding baseline methods under two metrics. We draw the following conclusions: Compared to all baseline models, our proposed FedGCV achieves the optimal performance across four datasets (except in the ideal retrain



(a) Cora (b) CiteSeer

(c) PubMed (d) Tolokers

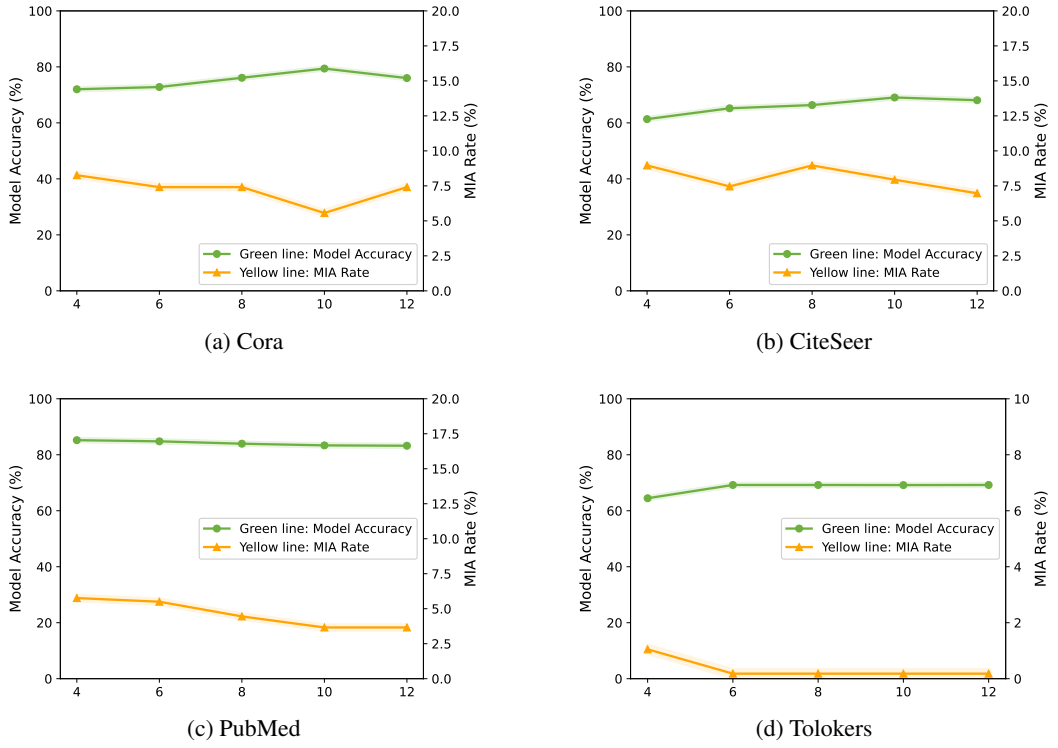
Figure 1: Unlearning degree and model accuracy are measured using four datasets: Cora, CiteSeer, PubMed, and Tolokers. The bar charts represent Model Accuracy, where higher performance is better; the line charts represent the MIA Rate, where lower values indicate higher unlearning levels.

scenario), and in many cases surpasses the performance of retrain. Notably, FedGCV achieves over 90% unlearning efficiency on the Cora, CiteSeer, PubMed, and Tolokers datasets, second only to the unlearning MIA rate of retraining. Furthermore, our framework compensates for model performance loss to over 90% of the pre-unlearning accuracy, making it the best among all baseline methods. Even on heterophilic subgraph dataset like Tolokers, it still performs exceptionally well. Compared with the state-of-the-art approach, ReGEnUnlearn Liu & Liu (2025), our proposed method simultaneously improves utility by 5.21% and reduces privacy leakage by 2.00%, on average across four datasets, demonstrating a favorable balance between performance and privacy.

4.3 Ablation Studies

To evaluate the effectiveness of each module in FedGCV, we conducted ablation studies across four datasets (e.g., Cora, CiteSeer, PubMed and Tolokers) and employed two evaluation metrics. Specifically, we denote simplified approaches achieved by removing "GRU-based gradient correction for unlearning" and "virtual client compensation for model utility" as "w/o GRU" and "w/o Virtual Client", respectively. It can be observed in Figure 1 that the overall performance declines whenever any component is removed, indicating each component makes significant contributions.

For instance, when the GRU was disabled and replaced with a traditional gradient ascent strategy in models trained on the Cora dataset, the MIA rate reached 24.07%, an increase of nearly 20% compared to when it was enabled. In addition, when the Virtual Client was disabled, model accuracy dropped from 82.42% to 65.37%. However, after incorporating the Virtual Client, accuracy reached 79.41%, clearly enhancing model utility.



(a) Cora (b) CiteSeer

(c) PubMed (d) Tolokers

Figure 2: Precision curves, MIA member recognition rates, and their standard deviation bands for different τ values on the Cora, CiteSeer, PubMed, and Tolokers datasets.

4.4 Sensitivity Analysis on Hyperparameters

Here, we conduct a detailed sensitivity analysis of the hyperparameters involved in the proposed FedGCV. Our FedGCV includes a hyperparameter τ that defines the maximum permissible norm for model parameter updates. Update steps exceeding τ are scaled proportionally to remain within the τ threshold. We plotted the MIA rate metric and model accuracy curves across four datasets, demonstrating performance under different hyperparameter values. As shown in Figure 2, performance fluctuates minimally across varying hyperparameter values, validating that FedGCV exhibits low sensitivity to these parameters as long as they are within reasonable ranges.

5 Conclusion

This paper investigates the issue of graph federated unlearning to address data privacy and the right to be forgotten. We propose a novel graph federated unlearning framework, graph **Fed**erated unlearning method with **G**radient **C**orrection and **V**irtual client (FedGCV), which balances model utility while ensuring effective unlearning. Specifically, FedGCV first introduces a gradient rectification unlearning mechanism to correct the direction of forgotten gradients. This minimizes interference with the learning performance of retaining clients while weakening the influence of forgotten clients. Subsequently, we design a VGAE-based virtual client generation strategy to synthesize virtual subgraphs that retain the original client's subgraph node feature distributions and graph spectral information without exposing raw data. These virtual subgraphs maintain similarity to real data in node feature distributions, Laplacian spectral properties, and propagation structures while preserving privacy, enabling effective participation in federated training to compensate for performance loss caused by unlearning. Extensive experimental results demonstrate that FedGCV significantly reduces the risk of member inference attacks while stably maintaining the model's predictive performance on retained data. Compared with various existing federated unlearning methods, FedGCV achieves a superior balance between unlearning effectiveness and model utility, validating its effectiveness and practicality in graph federated unlearning scenarios.

AI USE STATEMENT

We used large language model tools to assist with language editing, literature discovery, and checking the clarity of mathematical derivations. The authors independently verified the technical arguments, citations, and experimental claims and take full responsibility for the accuracy, originality, and integrity of the manuscript.

ETHICS STATEMENT

This work studies graph federated unlearning, with the aim of reducing the influence of data subject to a deletion request while preserving utility on retained data. The proposed virtual-client mechanism may create artifacts derived from the data to be forgotten, including spectral information, feature statistics, generator parameters, and synthetic graphs. Such artifacts could retain or reveal sensitive information even when raw data are not shared. Membership inference results alone do not establish privacy guarantees or compliance with a right-to-be-forgotten request. Deployments must define the scope of deletion, control access to derived artifacts, and assess leakage from both the released model and any retained artifacts. Our experiments use established research datasets; they do not establish the safety of deployment on sensitive personal data.

REPRODUCIBILITY STATEMENT

We evaluate the method on publicly available graph datasets and describe the model architectures, data partitioning, baselines, evaluation metrics, training configurations, and hyperparameters in the paper and appendix. We report the computational environment and random seeds to support independent replication. The implementation and experiment scripts are available in the anonymous repository linked in the abstract.

REFERENCES

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *ACM SIGSAC Conference on Computer and Communications Security*, pp. 308–318, 2016.

Jing Bai, Wentao Yu, Zhu Xiao, Vincent Havyarimana, Amelia C. Regan, Hongbo Jiang, and Licheng Jiao. Two-stream spatial–temporal graph convolutional networks for driver drowsiness detection. *IEEE Transactions on Cybernetics*, 52(12):13821–13833, 2022.

Deyu Bo, Xiao Wang, Yang Liu, Yuan Fang, Yawen Li, and Chuan Shi. A survey on spectral graph neural networks. *arXiv preprint arXiv:2302.05631*, pp. 1–9, 2023.

Rob Bonta. California consumer privacy act (CCPA). *Retrieved from State of California Department of Justice: https://oag. ca. gov/privacy/ccpa*, pp. 1–12, 2022.

Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *International Conference on Computational Statistics*, 2010. URL https://api.semanticscholar.org/CorpusID:115963355.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *IEEE Symposium on Security and Privacy*, pp. 141–159, 2021.

Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *IEEE Symposium on Security and Privacy*, pp. 463–480, 2015.

Min Chen, Zhikun Zhang, Tianhao Wang, Michael Backes, Mathias Humbert, and Yang Zhang. When machine unlearning jeopardizes privacy. In *ACM SIGSAC Conference on Computer and Communications Security*, pp. 896–911, 2021.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, pp. 1–15, 2014.

Rishav Chourasia and Neil Shah. Forget unlearning: Towards true data-deletion in machine learning. In *International Conference on Machine Learning*, pp. 6028–6073, 2023.

Gabriele Corso, Hannes Stark, Stefanie Jegelka, Tommi Jaakkola, and Regina Barzilay. Graph neural networks. *Nature Reviews Methods Primers*, 4(1):17, 2024.

Ayush Dave and Animesh Kumar Agrawal. A comparative study with GDPR, HIPAA, CCPA, PIPEDA and DPDPA. In *IEEE International Conference on Computer, Electronics, Electrical Engineering & their Applications*, pp. 1–6, 2025.

Jéssica Batista de Oliveira, Eduarda Stephanie de Oliveira Nascimento, Eneias Pinheiro de Oliveira Júnior, and Weliton do Nascimento Alexandre. A proteção de dados pessoais e a aplicação da lgpd no brasil. *NATIVA-Revista de Ciências, Tecnologia e Inovação*, 7(1):166–178, 2025.

Chongyu Fan, Jiancheng Liu, Yihua Zhang, Eric Wong, Dennis Wei, and Sijia Liu. Salun: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. *arXiv preprint arXiv:2310.12508*, pp. 1–31, 2023.

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9304–9312, 2020.

Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens Van Der Maaten. Certified data removal from machine learning models. In *International Conference on Machine Learning*, pp. 3832–3842, 2020.

Anisa Halimi, Swanand Kadhe, Ambrish Rawat, and Nathalie Baracaldo. Federated unlearning: How to efficiently erase a client in FL? *arXiv preprint arXiv:2207.05521*, pp. 1–13, 2022.

Zecheng He, Tianwei Zhang, and Ruby B Lee. Model inversion attacks against collaborative inference. In *Annual Computer Security Applications Conference*, pp. 148–162, 2019.

Zhehao Huang, Xinwen Cheng, JingHao Zheng, Haoran Wang, Zhengbao He, Tao Li, and Xiaolin Huang. Unified gradient-based machine unlearning with remain geometry enhancement. In *Advances in Neural Information Processing Systems*, pp. 26377–26414, 2024.

George Karypis and Vipin Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20(1):359–392, 1998.

Thomas N Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, pp. 1–3, 2016.

Hyeyoung Ko, Suyeon Lee, Yoonseo Park, and Anna Choi. A survey of recommendation systems: Recommendation models, techniques, and application fields. *Electronics*, 11(1):1–48, 2022.

Kemou Li, Qizhou Wang, Yue Wang, Fengpeng Li, Jun Liu, Bo Han, and Jiantao Zhou. LLM unlearning with LLM beliefs. *arXiv preprint arXiv:2510.19422*, pp. 1–33, 2025.

Li Li, Yuxi Fan, Mike Tse, and Kuo-Yi Lin. A review of applications in federated learning. *Computers & Industrial Engineering*, 149:1–58, 2020.

Maoxi Li, Mengying Shu, and Tianyu Lu. Anomaly pattern detection in high-frequency trading using graph neural networks. *Journal of Industrial Engineering and Applied Science*, 2(6):77–85, 2024.

Bo Liu, Ming Ding, Sina Shaham, Wenny Rahayu, Farhad Farokhi, and Zihuai Lin. When machine learning meets privacy: A survey and outlook. *ACM Computing Surveys*, 54(2):1–36, 2021.

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

Sijia Liu, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Yuguang Yao, Chris Yuhao Liu, Xiaojun Xu, Hang Li, et al. Rethinking machine unlearning for large language models. *Nature Machine Intelligence*, pp. 1–14, 2025.

Sitao Luan, Chenqing Hua, Qincheng Lu, Jiaqi Zhu, Mingde Zhao, Shuyuan Zhang, Xiao-Wen Chang, and Doina Precup. Revisiting heterophily for graph neural networks. *Advances in Neural Information Processing Systems*, 35:1362–1375, 2022.

Zhongtian Ma, Qiaosheng Zhang, Bocheng Zhou, Yexin Zhang, Shuyue Hu, and Zhen Wang. Graph attention is not always beneficial: A theoretical analysis of graph attention mechanisms via contextual stochastic block models. In *International Conference on Machine Learning*, pp. 1–41, 2025.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *International Conference on Artificial Intelligence and Statistics*, pp. 1273–1282, 2017.

Thanh Tam Nguyen, Thanh Trung Huynh, Zhao Ren, Phi Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen. A survey of machine unlearning, 2024. URL https://arxiv.org/abs/2209.02299.

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? In *International Conference on Learning Representations*, pp. 1–15, 2023.

Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, Yann Ollivier, and Hervé Jégou. White-box vs black-box: Bayes optimal strategies for membership inference. In *International Conference on Machine Learning*, pp. 5558–5567, 2019.

Ayush K Tarun, Vikram S Chundawat, Murari Mandal, and Mohan Kankanhalli. Fast yet effective machine unlearning. *IEEE Transactions on Neural Networks and Learning Systems*, 35(9): 13046–13055, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30:1–11, 2017.

Paul Voigt and Axel Von dem Bussche. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 1(1):1–88, 2016.

Qizhou Wang, Jin Peng Zhou, Zhanke Zhou, Saebyeol Shin, Bo Han, and Kilian Q Weinberger. Rethinking LLM unlearning objectives: A gradient perspective and go beyond. In *International Conference on Learning Representations*, pp. 1–35, 2025a.

Yue Wang, Qizhou Wang, Feng Liu, Wei Huang, Yali Du, Xiaojiang Du, and Bo Han. Gru: Mitigating the trade-off between unlearning and retention for large language models. In *International Conference on Machine Learning*, pp. 1–21, 2025b.

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.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S Yu. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24, 2020.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, pp. 1–17, 2019.

Zhilin Yang, William Cohen, and Ruslan Salakhudinov. Revisiting semi-supervised learning with graph embeddings. In *International Conference on Machine Learning*, pp. 40–48, 2016.

Wentao Yu. Homophily heterogeneity matters in graph federated learning: A spectrum sharing and complementing perspective. *arXiv:2502.13732*, pp. 1–15, 2025. URL https://arxiv.org/abs/2502.13732.

Wentao Yu, Sheng Wan, Guangyu Li, Jian Yang, and Chen Gong. Hyperspectral image classification with contrastive graph convolutional network. *IEEE Transactions on Geoscience and Remote Sensing*, 61(1):1–15, 2023.

Wentao Yu, Shuo Chen, Yongxin Tong, Tianlong Gu, and Chen Gong. Modeling inter-intra heterogeneity for graph federated learning. In *AAAI Conference on Artificial Intelligence*, pp. 22236–22244, 2025a.

Wentao Yu, Chen Gong, Bo Han, Lixin Fan, and Qiang Yang. Integrating commonality and individuality for graph federated learning: A graph spectrum perspective. *Authorea Preprints*, pp. 1–16, 2025b. URL https://www.techrxiv.org/doi/full/10.36227/techrxiv.175502607.78991557.

Wentao Yu, Shuo Chen, Chen Gong, Bo Han, Gang Niu, and Masashi Sugiyama. Atom-motif contrastive transformer for molecular property prediction. *ACM Transactions on Intelligent Systems and Technology*, 1(1):1–28, 2026.

Haoyu Zhang, Yuxuan Cheng, Wenqi Fan, Yulong Chen, and Yifan Zhang. *Rethinking Graph Domain Adaptation: A Spectral Contrastive Perspective*, pp. 448–464. Springer Nature Switzerland, October 2025. ISBN 9783032061065. doi: 10.1007/978-3-032-06106-5_26. URL http://dx.doi.org/10.1007/978-3-032-06106-5_26.

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

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning. *arXiv preprint arXiv:2404.05868*, pp. 1–32, 2024.

Hang Zhou, Wentao Yu, Sheng Wan, Yongxin Tong, Tianlong Gu, and Chen Gong. Traffic pattern sharing for federated traffic flow prediction with personalization. In *International Conference on Data Mining*, pp. 1–10, 2024.

Hang Zhou, Wentao Yu, Sheng Wan, Yongxin Tong, Tianlong Gu, and Chen Gong. FedTPS: traffic pattern sharing for personalized federated traffic flow prediction. *Knowledge and Information Systems*, 1(1):1–27, 2025.

A Appendix

A.1 Implementation Details

A.1.1 Datasets

In this experiment, we employ the citation-network benchmarks Cora, CiteSeer, and PubMed (Yang et al., 2016), and the Tolokers graph (Platonov et al., 2023). The statistics below describe the original graphs before any client partition.

Cora: The Cora dataset is an academic paper citation network comprising 2,708 nodes, each representing a paper. Each paper's features are represented by a 1,433-dimensional bag-of-words vector, reflecting the presence of specific terms within the paper. The dataset contains 5,429 citation edges, typically processed as an undirected graph. Papers are categorized into 7 topics, including neural networks and genetic algorithms, primarily for node classification tasks.

CiteSeer: The CiteSeer dataset is another academic paper citation network, slightly larger than Cora with 3,327 nodes and 4,732 citation edges. Each paper is represented as a 3,703-dimensional sparse binary vector and categorized into 6 domains, such as artificial intelligence and machine learning. This dataset is frequently used for node classification and graph representation learning tasks, serving as a classic benchmark for graph neural network validation.

PubMed: The PubMed dataset originates from the biomedical literature citation network, comprising 19,717 nodes and 44,338 edges. Each paper's feature is a 500-dimensional TF-IDF vector representing word importance within the paper. There are 3 categories, such as diabetes-related

topics. Due to its large scale and sparse real-valued features, PubMed is frequently used for semi-supervised node classification and testing the performance of graph convolutional networks.

Tolokers: Tolokers represents workers on a crowdsourcing platform, with an edge between workers who annotated the same task (Platonov et al., 2023). The original graph has 11,758 nodes, 519,000 edges, and 10-dimensional node features under the source dataset's edge-count convention. We use it for node classification on a graph with different structural properties from the citation networks.

A.1.2 BASELINES

We consider the following joint federated unlearning methods:

(1) Retraining from scratch: This method retrains an initialized model using data from the retaining clients, representing the ideal state that unlearning tasks aim to approximate.

(2) Projected Gradient Ascent(PGA) Halimi et al. (2022): Projection Gradient Ascent (PGA): PGA is an unlearning method designed to maximize local client loss within a controlled range.

(3) Elastic Weight Consolidation-Stochastic Gradient Ascent (EWC-SGA) Wu et al. (2022): EWC-SGA combines stochastic gradient ascent on the removed client loss with elastic weight consolidation to restrict updates on parameters that are important to the remaining clients to remove target clients without requiring full model retraining.

(4) Noisy Gradient Descent (Noisy-GD) Chourasia & Shah (2023): Noisy-GD injects random noise into gradient descent updates to reduce the influence of removed data by perturbing model parameters.

(5) ULKD: We follow the baseline description in Liu & Liu (2025); ULKD is a server-side storage history method that uses distillation techniques to eliminate target client data-sharing while enhancing model utility.

(6) SGA-Random (our baseline, implemented using stochastic gradient updates (Bottou, 2010)): SGA-Random performs stochastic gradient ascent using randomly sampled data from the removed client to erase its contribution from the model.

(7) SGA-Degree: SGA-Degree applies stochastic gradient ascent with degree-aware sampling, prioritizing nodes with higher structural influence during unlearning.

(8) ReGEnUnlearn Liu & Liu (2025): ReGEnUnlearn optimizes sampling strategies that minimize cross-client subgraph interference through Reinforcement learning-driven Federated Policy Sampler, while integrating a Parameter-free Graph Prompt Knowledge Distillation module to achieve precise retention of target client's unique structural knowledge and gradient-ascending complete unlearning.

A.1.3 MIA EVALUATION METRIC

In our implementation, MAIT evaluates unlearning via a membership inference attack (MIA) that uses the per-sample loss as the membership statistic, and reports the **MIA rate** on the target data which should be forgotten. To make the pre/post comparison well-defined, we fix the decision threshold at the value fitted on the pre-unlearning model and reuse it for all post-unlearning evaluations.

Let $\mathcal{D}_{\mathrm{tgt}}$ denote the target dataset. Let $\mathcal{D}_{\mathrm{ret}}$ denote a retained dataset drawn from other clients. Let $\mathcal{D}_{\mathrm{vir}}$ denote a virtual/non-member dataset generated from $\mathcal{D}_{\mathrm{ret}}$.

Let $\boldsymbol{\theta}$ be the parameters of the model being evaluated, $f(\cdot;\boldsymbol{\theta})$ the model, and $\mathcal{L}(\cdot,\cdot)$ the task loss (cross-entropy). For each sample $(\mathbf{x}, y)$, define the membership statistic as the sample loss:

$$\ell(\mathbf{x}, y; \boldsymbol{\theta}) := \mathcal{L}(f(\mathbf{x}; \boldsymbol{\theta}), y)\,.$$

Let $\boldsymbol{\theta}^{\mathrm{pre}}$ be the pre-unlearning model. MAIT fits a decision threshold τ_{pre} using $\boldsymbol{\theta}^{\mathrm{pre}}$ together with member/non-member reference data. Intuitively, τ_{pre} is chosen so that losses of members (from $\mathcal{D}_{\mathrm{tgt}}$) are more likely to fall below the threshold than losses of non-members (from $\mathcal{D}_{\mathrm{vir}}$), enabling membership discrimination.

Let $\mathbb{I}[\cdot]$ be the indicator function. Given the fixed threshold τ_{pre}, MAIT predicts a sample as a member if its loss is smaller than τ_{pre}. The reported MIA rate is then the fraction of target samples classified as members:

$$\mathrm{MIA}(\boldsymbol{\theta}; \tau_{\mathrm{pre}}) = \frac{1}{|\mathcal{D}_{\mathrm{tgt}}|} \sum_{(\mathbf{x},y)\in\mathcal{D}_{\mathrm{tgt}}} \mathbb{I}[\ell(\mathbf{x}, y; \boldsymbol{\theta}) < \tau_{\mathrm{pre}}] \,.$$

A lower $\mathrm{MIA}(\boldsymbol{\theta}; \tau_{\mathrm{pre}})$ indicates that the target data becomes harder to distinguish as training members and better unlearning.

A.2 Limitations

While our proposed FedGCV framework demonstrates strong empirical performance in graph federated unlearning, several limitations remain and some aspects can be further improved within standard assumptions.

Computational overhead. FedGCV introduces marginal additional computation from gradient correction and virtual client construction, especially for spectral decomposition and VGAE based generation on large graphs. Nevertheless, the overall overhead remains moderate in practice (Appendix A.4.1).

Approximation assumptions. Our method is derived under a local first order approximation, which mainly captures local interactions between unlearning and retention objectives. Although higher order effects are not explicitly modeled, such approximations are widely used in non-convex optimization and perform well in our experiments.

Applicability to broader settings. Our current framework focuses on static graph federated learning. Extending it to more complex scenarios, such as dynamic graphs, streaming data, or asynchronous federated optimization, is non-trivial and remains for future work.

A.3 Theoretical Justification and Methodology

A.3.1 Detailed Introduction of RTBF

In the literature of GFL, RTBF refers to the requirement that when a user withdraws from the system, the central server and all other clients should remove the message learned from that user. This helps better preserve user privacy and prevents malicious clients from misusing such information. Formally, consider a trained global model $\hat{\mathbf{w}}^*$ that approximates the solution of Eq. equation 2 and suppose the j-th client with local graph data $\mathcal{G}_j$ requests to be unlearned. Let $\mathrm{EXT}_j(\mathbf{w}; \mathcal{G}, \mathbf{X}, \mathbf{Y})$ denote an extraction function that measures the degree of information about $\mathcal{G}_j$ that can be inferred directly from $\mathbf{w}$ and indirectly from a set of local graph data $(\mathcal{G}, \mathbf{X}, \mathbf{Y})$. The goal of RTBF in GFL is to update the original model parameters from $\hat{\mathbf{w}}^*$ to $\hat{\mathbf{w}}^{\mathrm{u}}$ such that

$$\begin{aligned} &\mathrm{EXT}_j\left(\hat{\mathbf{w}}^{\mathrm{u}}; \{\mathcal{G}_i, \mathbf{X}_i, \mathbf{Y}_i\}_{i=1}^{M} \backslash \{\mathcal{G}_j, \mathbf{X}_j, \mathbf{Y}_j\}\right) \\ &\qquad \ll \mathrm{EXT}_j\left(\hat{\mathbf{w}}^*; \{\mathcal{G}_i, \mathbf{X}_i, \mathbf{Y}_i\}_{i=1}^{M}\right), \end{aligned} \tag{17}$$

i.e., the message about $(\mathcal{G}_i, \mathbf{X}_i, \mathbf{Y}_i)$ that can be extracted is notably reduced. This conceptual definition is adapted from Golatkar et al. (2020), further adjusted for GFL.

At first glance, it may seems that GFL is well suited for RTBF, as we can straightforwardly remove a client once the corresponding user raises a removal request. However, during federated learning, the user information has already been encoded into model parameters, shared with the central server via FedAvg as in Eq. equation 2, and then distributed to other clients. By membership inference attacks Liu et al. (2021) or model inversion attacks He et al. (2019), the user message may still be at risk of being extracted by others, even after the user has withdrawn from the FGL system.

A.3.2 Local Analysis of Gradient Correction

Let L_r be the retained objective with L-Lipschitz gradient near $\mathbf{w}$. For $\mathbf{w}^+ = \mathbf{w} + \eta\widetilde{\mathbf{a}}_{\mathrm{u}}$, the descent lemma gives

$$L_r(\mathbf{w}^+) - L_r(\mathbf{w}) \le \eta\langle\nabla L_r(\mathbf{w}), \widetilde{\mathbf{a}}_{\mathrm{u}}\rangle + \frac{L\eta^2}{2}\|\widetilde{\mathbf{a}}_{\mathrm{u}}\|_2^2. \tag{18}$$

If $\langle \nabla L_r, \widetilde{\mathbf{a}}_{\mathrm{u}} \rangle \leq 0$, the first-order change in retained loss is nonpositive. This local bound establishes neither deletion nor closeness to a model retrained without the target. In particular, ℓ_2 regularization does not generally make the Hessian of a nonconvex GNN positive definite.

A.3.3 VIRTUAL-CLIENT ARTIFACTS AND DELETION SCOPE

The VGAE parameters, feature moments, eigenspaces, generated graphs, checkpoints, and updates may all depend on withdrawn data. Spectral truncation and a VGAE KL penalty do not imply differential privacy or a bound on reconstruction risk. When the deletion request covers all derived artifacts, these objects must be discarded, and any utility-repair data must be generated solely from retained data. Such a changed pipeline needs new experiments. If artifacts derived from withdrawn data are retained, the evaluation addresses a narrower, model-level empirical forgetting setting and must test attacks against accessible artifacts as well as the released model.

A.3.4 GENERALIZATION OF VIRTUAL CLIENT ON HETEROGENEOUS GRAPH DATA

Our method performs exceptionally well on heterogeneous graph data and demonstrates excellent generalization capabilities. The reason is that unlike traditional feature generation, we use the low-frequency eigenvectors U_k of the graph Laplacian (Eq. equation 13) to guide the generation. Since the low-frequency components capture the global connectivity and smooth structure of the graph Zhang et al. (2025) this ensures that the synthesized graph retains the core topological skeleton of the original subgraph even in a non-IID setting.

Furthermore, VGAE learns latent distribution statistics (mean and variance) rather than simply mimicking samples. This generative logic inherently possesses the ability to smooth heterogeneous noise.

A.4 DETAILED EXPERIMENTAL RESULTS

We conduct a series of supplementary experiments to further validate the practicality, generalizability, and optimality of our method.

A.4.1 COMPUTATIONAL OVERHEAD COMPARISON

We evaluate the computational efficiency of FedGCV against full retraining. The results demonstrate a significant speedup in unlearning time.

Table 2: Computational overhead comparison between FedGCV and Retraining.

Metric	**FedGCV**	**Retraining**
Time Cost (s)	2.47	57.04
Memory Usage (MiB)	25.34	18.14

Compared to standard retraining which takes 57.04s, FedGCV achieves a dramatic reduction in time overhead, dropping to only 2.47s (containing GRU and generation of virtual client). FedGCV incurs a slightly higher memory footprint. This extra overhead is primarily due to storing the reference gradients and the orthogonalization matrix during the correction step. However, this is a one-time cost that is well within the capacity of modern servers.

Furthermore, the server-side overhead of gradient correction is negligible and scales efficiently. The projection Eq.equation 10 involves only two aggregated vectors (Δ_u and Δ_r). Its complexity is $O(d)$ (d is the parameter count), which is independent of the number of clients M. Compared to local training costs ($O(N \cdot d)$), a single dot product and vector subtraction at the server takes negligible time. This $O(d)$ operation avoids full model retraining, significantly reducing total communication rounds and energy consumption. Therefore, FedGCV is very cost-effective in terms of both time and storage.

Table 3: Results with GAT and GraphSAGE backbones. Each entry shows the value before and after unlearning.

Backbone	Dataset	Accuracy ↑	MIA Rate ↓
GAT	Cora	80.90% → 77.96%	83.33% → 9.26%
	CiteSeer	70.89% → 65.32%	96.83% → 12.70%
	PubMed	85.24% → 80.50%	100.00% → 1.31%
	Tolokers	78.95% → 76.61%	92.41% → 6.24%
GraphSAGE	Cora	74.69% → 72.08%	92.59% → 3.70%
	CiteSeer	69.41% → 68.25%	95.24% → 9.52%
	PubMed	86.53% → 84.61%	97.12% → 0.52%
	Tolokers	78.20% → 76.61%	61.55% → 9.95%

Table 4: Results with 30 clients. Each entry shows the value before and after unlearning.

Dataset	Accuracy ↑	MIA Rate ↓
Cora	58.35% → 47.55%	100.00% → 5.56%
CiteSeer	63.01% → 62.93%	88.89% → 14.29%
PubMed	83.71% → 77.04%	98.95% → 3.93%
Tolokers	77.95% → 73.36%	98.82% → 3.54%

A.4.2 Backbone Robustness

Sufficiently exploring the robustness of our method across different backbone GNN architectures is necessary. Therefore, we have conducted extensive additional experiments using **GAT** and **GraphSAGE** as alternative backbones across all four datasets (Cora, CiteSeer, PubMed, and Tolokers). The result is displayed in Table 3

The utility performance maintains a good standard while MIA Rate declines sharply, indicating our method works across different backbones. It turns out FedGCV consistently achieves high unlearning effectiveness and utility across different architectures.

A.4.3 Scaling to More Clients

To further validate the applicability of our method, we evaluate FedGCV under larger scales on four datasets by increasing the number of clients to 30.

As shown in Table 4 , FedGCV maintains stable utility and significantly reduces MIA risk across all settings, demonstrating good scalability to more clients. The results show that even with more clients, FedGCV effectively unlearns target data while maintaining reasonable utility, confirming its scalability.

A.4.4 More Baselines

Retraining for a small number of additional federated rounds seems to be a better method compared to retraining for a full 30 epochs, which clearly requires less time and memory. Therefore, we conduct more experiments to demonstrate that our method is superior to the alternative and is unreplaceable. The result is shown in Table 5.

Without specific erasure mechanisms, the residual influence of the forgotten client remains embedded in the model's weights. Our experiments show that such alternatives fail to achieve a satisfactory MIA rate compared to FedGCV, as they cannot effectively zero out the specific structural features learned from the deleted graph nodes.

A.5 Notation Used in This Paper

We use the following symbols throughout the main text and appendix. Unless stated otherwise, j denotes the client requesting deletion; this is a client-level request in the present formulation.

Table 5: Experimental results comparing few-round retraining and FedGCV.

Dataset	Method	Accuracy ↑	MIA Rate ↓	Accuracy before unlearning	MIA Rate before unlearning
Cora	Few-round Retrain	55.99%	84.44%	82.42%	87.04%
	FedGCV (Ours)	**79.41%**	**5.56%**		
CiteSeer	Few-round Retrain	65.59%	85.24%	70.14%	88.89%
	FedGCV (Ours)	**69.07%**	**7.94%**		
PubMed	Few-round Retrain	**85.81%**	97.91%	86.52%	97.91%
	FedGCV (Ours)	83.35%	**3.66%**		
Tolokers	Few-round Retrain	**76.69%**	87.69%	78.10%	96.32%
	FedGCV (Ours)	69.17%	**0.18%**		

Table 6: Graph federated unlearning notation.

Symbol	Meaning
M, i, j	Number of clients, client index, and deletion-requesting client.
$\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i)$, $\mathbf{A}_i$	Client i's local graph, node and edge sets, and adjacency matrix.
$\mathbf{X}_i$, $\mathbf{Y}_i$	Node features and task labels at client i.
$\mathcal{D}_{\mathrm{u}}$, $\mathcal{D}_{\mathrm{r}}$	Data in the deletion request and data retained after deletion.
$\mathbf{w}^{(t)}$, $\mathbf{w}_i^{(t+1)}$	Global model at round t and client i's locally trained model.
$\hat{\mathbf{w}}^*$, $\hat{\mathbf{w}}^{\mathrm{u}}$	Model before and after the unlearning procedure.
$\mathbf{w}_{\mathrm{retrain}}$	Reference model trained from scratch on retained data.
p_i, $\bar{p}_i$	Original FedAvg weight and retained-client normalized weight $p_i / \sum_{k \neq j} p_k$.
$\hat{\mathcal{L}}_{\mathrm{u}}$, $\hat{\mathcal{L}}_{\mathrm{r}}$	Empirical task losses on deletion-requested and retained data.
$\mathbf{a}_{\mathrm{u}}$, $\mathbf{g}_{\mathrm{r}}$	Target-loss ascent vector $\nabla \hat{\mathcal{L}}_{\mathrm{u}}$ and retained-loss gradient $\nabla \hat{\mathcal{L}}_{\mathrm{r}}$.
$\mathbf{d}_i$, $\mathbf{\Delta}_{\mathrm{r}}$	Client update $\mathbf{w}_i^{(t+1)} - \mathbf{w}^{(t)}$ and weighted retained update $\sum_{i \neq j} \bar{p}_i \mathbf{d}_i$.
$\mathbf{\Delta}_{\mathrm{u}}$, η, β	Actual target-loss ascent update, step size, and its aggregation weight.
$\mathbf{L}$, $\mathbf{U}_k$, $\mathbf{Z}_{\mathrm{proj}}$	Graph Laplacian, its selected eigenvectors, and projected latent representation.
α, τ_{pre}	Virtual-client participation weight and pre-unlearning MIA threshold.

A node-level request requires specifying the removed node, its incident edges, and the retained neighbors whose message-passing inputs change.