PAGE: A Unified Approach for Federated Graph Unlearning

Yuming Ai*[1], Xunkai Li*[1], Jiaqi Chao[2], Bowen Fan[1], Zhengyu Wu[1], Yinlin Zhu[3], Rong-Hua Li[1], Guoren Wang[1]

[1]Beijing Institute of Technology, Beijing, China
[2]Shandong University, Shandong, China
[3]Sun Yat-Sen University, Guangdong, China
3120251027@bit.edu.cn, cs.xunkai.li@gmail.com, 202300800028@mail.sdu.edu.cn, 3220241345@bit.edu.cn, jeremywzy96@outlook.com, zhuylin27@mail2.sysu.edu.cn, lironghuabit@126.com, wanggrbit@gmail.com

Abstract

Federated graph learning (FGL) is a distributive framework for graph representation learning that prioritizes privacy preservation. The right to be forgotten embodies the ethical principle of prioritizing user autonomy over data usage. In the context of FGL, upholding this right requires the method to remove specific entities and their associated knowledge within local subgraphs (Meta Unlearning) and the complete erasure of the entire client (Client Unlearning). We are the first to systematically define the above two unlearn requests in federated graph unlearning. Several studies have attempted to address this challenge, but key limitations persist: incomplete unlearning support and residual knowledge permeation. To this end, we propose a **P**rototype-guided **A**dversarial **G**raph **E**raser for universal federated graph unlearning (**PAGE**), the first unified federated graph unlearning framework that extend to comprehensive unlearning requests. For meta unlearning, we employ the prototype gradients guide initial local unlearn, while adversarial graphs eliminate residual knowledge across the influenced clients. For client unlearning, PAGE exclusively utilizes adversarial graph generation to purge a departed client's influence from the remaining participants. PAGE outperforms existing methods on 8 benchmark datasets. It improves prediction accuracy by 5.08% (client unlearn) and 1.50% (meta unlearn), with up to 11.84% gain on large-scale graphs. Furthermore, ablation studies confirm its efficacy as a plug-in for other meta unlearn methods, boosting prediction performance up to 4.49% and unlearning performance up to 7.22%.

Introduction

The growing adoption of graph neural networks (GNNs) (Wu et al. 2020) in domains like social networks (Wang et al. 2018), bioinformatics (Qu et al. 2023), and recommendation systems (Cai et al. 2023) underscores the value of graph-structured data. However, real-world graphs are inherently distributed across clients and often contain sensitive information. Centralized graph learning methods, which require data aggregation, risk privacy breaches and violate regulations such as GDPR (Regulation 2018) and CCPA (Pardau 2018). Federated graph learning (He et al. 2021) thus emerges, combining privacy-preserving computation with graph representation learning. This approach ensures regulatory compliance while mitigating performance degradation from data heterogeneity via cross-client graph modeling.

One urgent privacy-preserving compliance to meet is granting users' rights to be forgotten (Yang et al. 2024). However, most existing federated graph learning methods lack this capability, limiting their applicability in real-world deployments and introducing model bias (Chang and Shokri 2023) and security vulnerabilities (Tolpegin et al. 2020). Unlike most existing Federated Unlearning methods (Romandini et al. 2024; Zhong et al. 2025; Zhao et al. 2024) that are primarily designed for independent and identically distributed (i.i.d.) non-graph data such as images or text, designing a unified framework for Federated Graph Unlearning (FGU) presents unique challenges.

Unlearning entities from graph-structured data requires not only removing the target node but also its connectivity with neighboring nodes, thereby disrupting the topological integrity. Moreover, due to the dependency of GNNs on information propagation, such structural disturbances can significantly degrade the quality of neighboring nodes' learned representations. In the context of FGU, this work formally proposes two unlearn requests that align with the collaborative training architecture:

◆ **Meta Unlearning:** Request to remove specific entities and their associated structural information from a client's local subgraph during the federated training process. This ensures that historical interactions of forgotten entities no longer contribute to cross-client knowledge transfer.

◆ **Client Unlearning:** Request to remove an entire client during the federated training process, ensuring that all associated entities and their interactions cease contributing to the global learning process. The knowledge originated from forgotten clients that infused to other models during the federated process should also be removed.

Formal mathematical representations of these two unlearn requests are provided in the next section. Both requests reflect common real-world demands for protecting user data privacy, such as requests to delete personal accounts or to fully withdraw a client from federated participation due to

*These authors contributed equally.

new regulatory restraints or compromised security.

Challenge. Although some prior work has attempted to address these challenges, key limitations persist:

① **Incomplete Unlearning Support**: Existing methods handle client unlearnin and meta unlearning in isolation, lacking a unified framework to coordinate multi-level unlearning. This limits their effectiveness in addressing heterogeneous unlearning requirements in practical settings.

② **Residual Knowledge Permeation**: During federated training, information from data marked for unlearning may propagate to other clients via gradient exchanges and graph-structural dependencies. Most prior approaches focus solely on the global model and the primary unlearning subject, neglecting cross-client contamination. Consequently, sensitive information might continue to be extracted.

Method. To address these limitations, we propose PAGE, a novel three-stage unlearning framework. The first stage, Prototype Matching for Local Unlearn, efficiently removes a client's unique contributions by projecting its data prototype onto a shared feature subspace and minimally fine-tuning the model toward this common representation. Subsequently, the second stage, Adversarial Graph Generation, validates the process by creating adversarial inputs engineered to maximize the output discrepancy between the original and unlearned models, thereby acting as sensitive probes for any residual knowledge. Finally, if knowledge permeation is detected, the third stage, Negative Knowledge Distillation for Influenced Unlearn, identifies the influenced clients via prototypical similarity and applies a targeted distillation to eradicate the infiltrated information, ensuring a comprehensive and robust unlearning outcome.

Contributions. (1) *New Perspective*. This paper establishes a unified theoretical framework for federated graph unlearning that supports multiple unlearn requests. It sets the standard for achieving precise decoupling between forgotten and retained knowledge in federated graph scenarios for the first time. (2) *New Method*. We propose **PAGE**, a novel approach comprising three key components: a prototype matching module for guiding local unlearning, an adversarial graph generation module for verifying unlearning and mitigating residual impact, and a negative knowledge distillation module for removing unintended knowledge transfer. (3) *SOTA Performance*. Across 8 benchmark datasets, PAGE achieves state-of-the-art performance, improving prediction accuracy by 5.08% in client unlearning and 1.50% in meta-unlearning scenarios. More impressively, it yields up to 11.84% improvement on large-scale graphs. PAGE also serves as an effective plug-in to enhance existing meta-unlearn methods as the ablation experiment reveals.

Preliminaries

Federated Graph Learning

In this subsection, we present a formal definition of federated graph learning (FGL). Assume that the FGL system comprises K clients $\mathcal{C} = \{c^1, c^2, ..., c^K\}$ and a central server. The global graph dataset is defined as $\mathcal{G} = \{\mathcal{G}^i | c^i \in \mathcal{C}\}$. The server retains no raw data but the global model parameter θ^g. Each client c^i holds a local subgraph $\mathcal{G}^i = \{\mathcal{V}^i, \mathcal{E}^i, \mathcal{X}^i\}$, where $\mathcal{V}^i$ represents the node set in c^i, $\mathcal{E}^i$ represents the edge set, $\mathcal{X}^i$ represents the feature set. In addition, each client holds a local model parameter θ^i with the same architecture as the global model. The objective of FGL is to collaboratively train the global graph neural network model $f_{\theta^g} : \mathcal{G} \to \mathcal{Y}$ (e.g., for node classification or link prediction) whose parameters θ^g are obtained by minimizing a global loss function:

$$\arg\min_{\theta^g} \mathcal{L}(\theta^g) = \sum_{i=1}^{K} \frac{|\mathcal{V}_i|}{|\mathcal{V}|} \cdot \mathcal{L}_i(\theta^g; \mathcal{G}^i) \tag{1}$$

For a more detailed exposition, we illustrate the training procedure of FGL at round t using **FedAvg** as an example.

Receive Message. Each client receives the global model parameters distributed by the server and initializes the local model through $\theta_t^i \leftarrow \theta_t^g$.

Local Update. Each client performs multiple rounds of local training using its own subgraph. The goal of local training is to obtain θ_{t+1}^i by optimizing the local loss function:

$$\theta_{t+1}^i = \arg\min_{\theta_t^i} \mathcal{L}_{task}(\theta_t^i; \mathcal{G}^i) \tag{2}$$

Upload Message. Each client uploads the local model θ_{t+1}^i and sampled data size n_i to the server.

Global Aggregation. The server collects the parameters uploaded by all K clients and performs weighted averaging:

$$\theta_{t+1}^g = \sum_{i=1}^{K} \frac{n_i}{\sum_j n_j} \cdot \theta_{t+1}^i \tag{3}$$

Federated Graph Unlearning

In this subsection, we give a formal definition of various unlearn requests and FGU. The unlearn requests are divided into two categories: meta unlearning and client unlearning.

Meta Unlearning. Meta unlearning represents the unlearning of the internal structure of the client subgraph. For example, in a social network scenario, if a user asks the platform to delete his account, the user node and all its associated edges need to be removed from the global social graph, and subsequent recommendation models no longer rely on their historical interactions. When client c^i receiving a meta unlearn request $\Delta\mathcal{G}^i = \{\Delta\mathcal{V}^i, \Delta\mathcal{E}^i, \Delta\mathcal{X}^i\}$, it can be decomposed into three dimensions of unlearning: node level $\Delta\mathcal{G}^i = \{\Delta\mathcal{V}^i, \varnothing, \varnothing\}$, edge level $\Delta\mathcal{G}^i = \{\varnothing, \Delta\mathcal{E}^i, \varnothing\}$ and feature level $\Delta\mathcal{G}^i = \{\varnothing, \varnothing, \Delta\mathcal{X}^i\}$. For each client, three dimensions of meta unlearn request may occur. Therefore, for the entire federated system, the total meta unlearn request is defined as $\Delta\mathcal{G} = \{\Delta\mathcal{V}, \Delta\mathcal{E}, \Delta\mathcal{X}\} = \bigcup_{i=1}^{K} \Delta\mathcal{G}^i$.

Client Unlearning. Client unlearning means that the entire client exits the FGL system, requiring the removal of all its data contributions. For instance, if a bank withdraws from collaboration due to policy restrictions, all related account nodes and associated transaction edges must be removed from the global transaction graph without affecting the performance of the risk control model of other banks. When receiving a client unlearn request $\Delta\mathcal{C} = \{c^{u_1}, c^{u_2}, ..., c^{u_M}\}$, the FGL system needs to completely and verifiably remove all data contributions from some specific clients.

Existing graph unlearning (GU) methods support meta unlearn requests: GIF (Wu et al. 2023a) and IDEA (Dong et al. 2024) quantify first-order deletion impacts via closed-form parameter adjustments, while D2DGN (Sinha, Mandal, and Kankanhalli 2024) and MEGU (Li et al. 2024) balance unlearning and reasoning through specialized loss functions. For knowledge graphs, FedLU (Zhu, Li, and Hu 2023) eliminates triplet knowledge with neuroscience-inspired global propagation, and FedDM (Liu and Fang 2024) leverages diffusion models to mitigate knowledge impacts via noisy data generation. Correspondingly, most federated unlearning methods only address client unlearn requests: FedEraser (Romandini et al. 2024) reconstructs models via historical update calibration; FUSED (Zhong et al. 2025) trains sparse adapters for knowledge overwriting; MoDe (Zhao et al. 2024) employs staged momentum degradation and distillation; ReGEnUnlearn (Liu and Liu 2025) reduces subgraph interference through optimal sampling and client-specific knowledge extraction.

Notions. Assume that $\hat{\theta}$ is the randomly initialized original parameters As defined above, the FGL system obtains model parameters based on the client set $\mathcal{C}$ and the graph dataset $\mathcal{G}$ by $\theta^o = FGL(\hat{\theta}; \mathcal{C}; \mathcal{G})$. When an unlearn request $< \Delta\mathcal{G}, \Delta\mathcal{C} >$ is received, the retrained model parameters θ^* are trained from scratch on retained graph $\mathcal{G}_r = \mathcal{G} \setminus \Delta\mathcal{G}$ and retained client set $\mathcal{C}_r = \mathcal{C} \setminus \Delta\mathcal{C}$ by $\theta^* = FGL(\hat{\theta}; \mathcal{C}_r; \mathcal{G}_r)$. The parameters of the unlearned model, denoted $\bar{\theta}$ are obtained by applying the FGU algorithm to the original model parameters θ^o, defined as: $\bar{\theta} = FGU(\theta^o; \mathcal{C}; \mathcal{G}; \Delta\mathcal{C}; \Delta\mathcal{G})$.

Objective. From a mathematical definition, the goal of the FGU algorithm is to minimize the difference between $\bar{\theta}$ and θ^*. From a practical standpoint, the goal of a FGU algorithm is to, under the strict requirements of privacy compliance, efficiently and completely eliminate the influence of specified data elements (nodes, edges, subgraphs, or entire clients) on the already-trained graph model, while simultaneously minimizing any degradation in the model's ability to represent the remaining data—thereby striking an optimal balance between unlearning and preservation.

Unlearning Verification. A membership inference attack is a privacy-stealing technique that aims to determine if a specific data record was part of a model's training set. The attacker accomplishes this by building an inference model that analyzes the target model's responses to distinguish between training and non-training data.

Proposed Method

In this section, we introduce PAGE, designed to address federated graph unlearning under multiple unlearn requests. The methodology comprises three core steps: (1) Prototype Matching for Local Unlearn: The server leverages prototypes to define unlearning objectives, guiding clients in the precise erasure of local knowledge; (2) Adversarial Graph Generation: Constructing negative samples implicitly containing knowledge for unlearning by maximizing model discrepancies before and after unlearning; (3) Negative Knowledge Distillation for Influence Unlearn: Eliminating the impact of knowledge permeation on associated clients.

Architecture Overview

As shown in Figure 1, in the preprocessing stage on the left, each client calculates a local prototype vector and uploads it to the server. The server calculates the cosine similarity between the vectors and filters out the influenced clients based on the similarity. On the right, PAGE operates in three stages. First, it performs precise local unlearning by leveraging semantic prototypes, enabling the server to isolate private data components based on client-provided feature centroids. Next, it validates the unlearning efficacy by generating adversarial graph samples that maximize the discrepancy between the model's pre- and post-unlearning states, thereby creating a sensitive probe for residual knowledge. Finally, the method addresses federated knowledge leakage by using prototype similarity to identify influenced clients and then applying negative knowledge distillation—guided by the adversarial samples—to approximate the desired unlearned state.

It is worth noting that under the client unlearn request, there is no need to perform prototype matching for local unlearning. When all of the data in the client need to be forgotten, the noise model $\hat{\theta}$, which is initialized randomly, represents the unlearned model $\bar{\theta}^u$. Adversarial graph generation module and negative knowledge distillation module together form the influenced unlearning component.

Prototype Matching

The topological correlations in graph data cause conventional parameter-level unlearning to disrupt structural semantics, while federated learning's global-local knowledge coupling compounds these difficulties. We thus propose a prototype matching-based local unlearning mechanism with core principles: (1) Representing data distributions abstractly via class prototype vectors (feature-space centroids), avoiding direct raw graph manipulation; (2) Generalizing gradient matching's behavioral constraint concept to feature-space prototype alignment, reducing computation costs while preserving graph structure modeling.

Prototype. The client c^i trains a graph neural network f_{θ^i} on the local dataset $\mathcal{G}^i$ and calculates its category prototype vector p^i. Taking the node classification task as an example, for category c, the prototype is defined as the mean representation of the nodes of this category in the feature space:

$$p_c^i = \frac{1}{|\mathcal{V}^i|} \sum_{(x^i, y^i = c)} f_{\theta^i}(x^i) \tag{4}$$

where $\mathcal{V}^i$ is the set of all nodes of the category c in the client c^i. Then, all clients upload their model parameters $\{\theta^i | i = 1, 2, ..., K\}$ and prototype sets $\mathcal{P}$ to the server.

$$\mathcal{P} = \{p_c^i | i = 1, 2, ..., K; c \in C\} \tag{5}$$

The client c^u computes the unlearn prototype p^{del} derived from the unlearn data $\Delta\mathcal{G}^u$:

$$p^{del} = \{\frac{1}{|\Delta\mathcal{V}^u|} \sum_{(x^u, y^u = c)} f_{\theta^u}(x^u) | c \in C\} \tag{6}$$

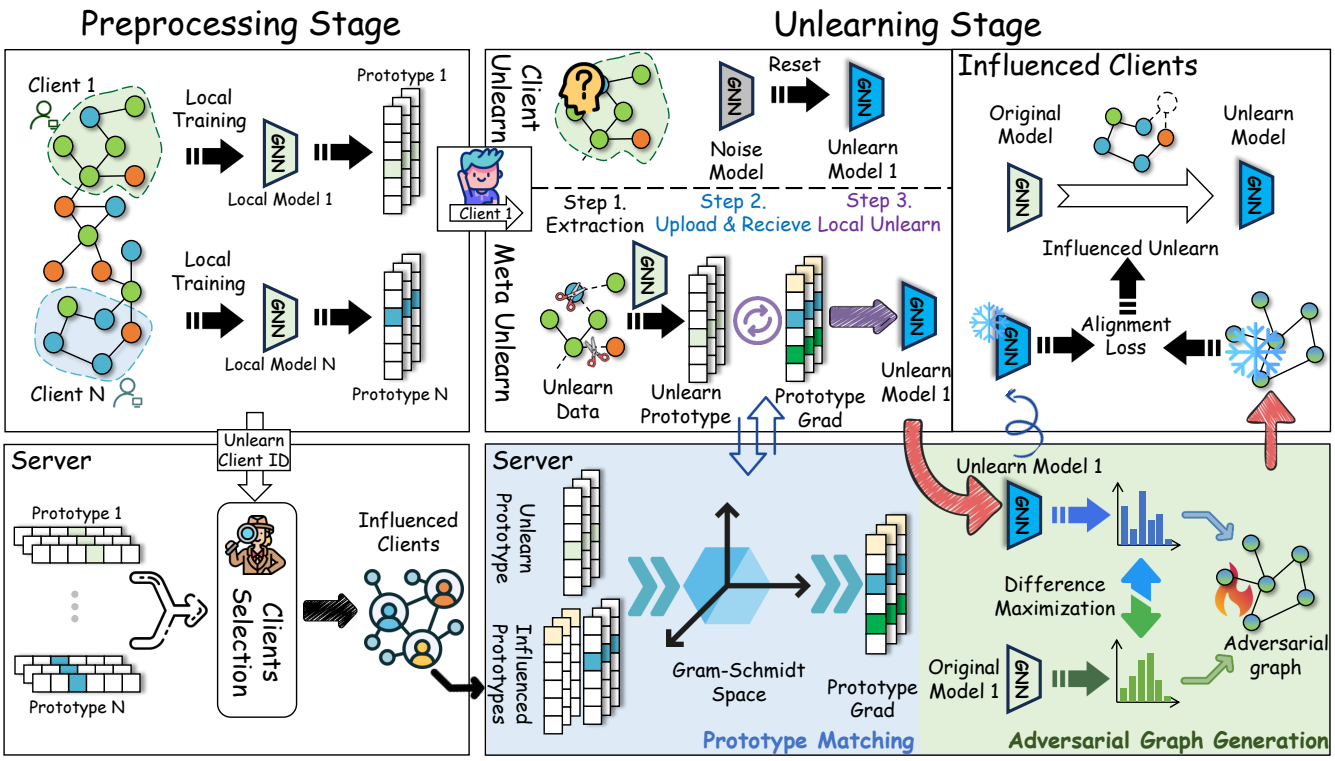

Figure 1: Overview of our proposed PAGE. Left: preprocessing before unlearning. Right: unlearning stage. The server performs prototype matching to guide the client to local unlearn. Based on the original model and unlearned model, adversarial graph samples are generated to guide the influenced client to perform influenced unlearn.

Gram-Schmidt Prototype Space. The server aggregates prototype sets $\mathcal{P}$ from all non-unlearning clients. To eliminate redundant information and construct an orthogonal basis space, Gram-Schmidt orthogonalization is employed on the prototype vectors $\mathcal{P}$: $v_1 = p_1/||p_1||_2, v_i = p_i - \sum_{j=1}^{i-1} < p_i, v_j > v_j, \tilde{v}_i = v_i/||v_i||_2$

Subsequently, we obtain the orthogonal basis matrix $V = [\tilde{v}_1, \tilde{v}_2, ..., \tilde{v}_M] \in \mathbb{R}^{d \times M}, M << N$, which spans the global prototype space $Span(\mathcal{P})$. Then, orthogonal projection decomposition is performed to project the unlearning prototypes p^{del} onto $Span(\mathcal{P})$.

$$p_{com} = VV^T p^{del} = \sum_{i=1}^{M} < p^{del}, \tilde{v}_i > \tilde{v}_i \tag{7}$$

We can obtain the prototype grad $p_{priv} = p_{com} - p^{del}$, which satisfies the orthogonality.

$$< p_{priv}, \tilde{v}_i >= 0, \forall i \in \{1, 2, ..., M\} \tag{8}$$

This prototype grad p_{priv} is termed the pure private knowledge guiding vector. Then, distribute p_{priv} to the client c^u to perform local unlearn.

Local Unlearn. The client c^u performs local unlearn based on the prototype grad p_{priv}:

$$\mathcal{L} = MSE(p^{del}, p_{priv}) \tag{9}$$

And then, we can get the unlearned model paramaters $\bar{\theta}^u$.

Adversarial Graph Generation

In federated graph unlearning, relying solely on local unlearn operations cannot adequately verify knowledge elimination effectiveness. The primary reasons are: 1. Concealment of Residual Knowledge: Unlearning may only be effective at specific decision boundaries, requiring sensitive samples to expose model behavioral changes; 2. Need for Cross-Client Impact Quantification: Knowledge permeation levels demand quantifiable assessment tools to provide precise unlearning guidance for other influenced clients.

To address this, we propose a difference-maximized graph adversarial generation method. Its core principle systematically explores regions where model behavior changes most significantly after unlearning by simultaneously perturbing graph structures and node features.

First, the server loads the original model parameters and unlearned model parameters $\theta^u, \bar{\theta}^u$ of the target unlearning client c^u to establish a dual-model comparison framework. The initial inputs for adversarial graph generation comprise the node feature matrix $X_{init} \in \mathbb{R}^{N \times d}$ and adjacency matrix $A_{init} \in \{0, 1\}^{N \times N}$, which are initialized randomly.

Subsequently, gradient ascent is employed to optimize perturbations in continuous space, with differentiable adjacency matrix weights A_{var} and feature matrix X_{var} defined as optimization variables. Refer to Algorithm 1 for the specific optimization process.

Algorithm 1: Adversarial Graph Generation.

Input: Initial graph $\mathcal{G}_{init} = (A_{init}, X_{init})$, models f_{θ^u}, $f_{\bar{\theta}^u}$, hyperparameters $\lambda = 0.1$, $\epsilon_x = 0.1$
Output: Adversarial graph $\mathcal{G}_{adv}$
Initialize optimization variables A_{var}, X_{var}
while not converged **do**
  $\tilde{A} \leftarrow \sigma(A_{var})$, $A_{sym} \leftarrow (\tilde{A} + \tilde{A}^T)/2$
  $\mathcal{G} \leftarrow (X_{var}, A_{sym})$
  $\mathcal{L}_{diff} \leftarrow \text{CE}(f_{\theta^u}(\mathcal{G}), f_{\bar{\theta}^u}(\mathcal{G}))$
  $\mathcal{L}_{reg} \leftarrow \|A_{sym} - A_{init}\|_1$
  $\mathcal{L}_{adv} \leftarrow \mathcal{L}_{diff} - \lambda \mathcal{L}_{reg}$
  Update A_{var}, X_{var}
  $X_{var} \leftarrow X_{init} + \text{clip}(X_{var} - X_{init}, -\epsilon_x, \epsilon_x)$
end while
Post-processing:
$\Delta A \leftarrow \text{TopK}(|A_{var} - A_{init}|), K = 5$
Construct A_{final}:

$$A_{final}[i,j] = \begin{cases} 1 - A_{init}[i,j] & \forall (i,j) \in \Delta A \\ A_{var}[i,j] & \text{otherwise} \end{cases}$$

$X_{final} \leftarrow X_{init} + \text{clip}(X_{var} - X_{init}, -\epsilon_x, \epsilon_x)$
$\mathcal{G}_{adv} \leftarrow (X_{final},\ \{(i,j) \mid A_{final}[i,j] > 0.5\})$
Return $\mathcal{G}_{adv}$

Negative Knowledge Distillation

To address the challenge of knowledge permeation, this module proposes a negative knowledge distillation mechanism. Using graph adversarial samples, it compels affected clients to conform to the post-unlearning state in sensitive regions, thereby achieving targeted elimination of knowledge permeation.

The server broadcasts the unlearned model parameters $\bar{\theta}^u$ and graph adversarial samples $\mathcal{G}_{adv}$ to other affected clients. The output of the unlearned model on these graph adversarial samples represents the negative knowledge requiring elimination. In addition, in each affected client, the local positive knowledge needs to be preserved is obtained from the local model parameters θ^i and local data $\mathcal{G}^i$. Through this module we can get unlearned model parameters of each client $\bar{\theta}^i$.

Distillation architecture. To preserve positive knowledge, we want to minimize the loss between $\bar{\theta}^i$ and θ^i for the local dataset $\mathcal{G}^i$:

$$\mathcal{L}_{pos} = \mathcal{L}_{task}(f_{\bar{\theta}^i}(\mathcal{G}^i), f_{\theta^i}(\mathcal{G}^i)) \tag{10}$$

Similarly, in order to unlearn the knowledge requiring elimination, we hope that the loss between $\bar{\theta}^i$ and $\bar{\theta}^u$ on $\mathcal{G}_{adv}$ is as small as possible:

$$\mathcal{L}_{neg} = \mathcal{L}(f_{\bar{\theta}^i}(\mathcal{G}_{adv}), f_{\bar{\theta}^u}(\mathcal{G}_{adv})) \tag{11}$$

where, $\mathcal{L}$ is the chosen distillation measure, which is the Mean Squared Error (MSE) loss function in this paper.

Finally, we can get the final objective function:

$$\mathcal{L} = \mathcal{L}_{pos} + \alpha \mathcal{L}_{neg} \tag{12}$$

where α is the regularization hyperparameter that balances the trade-off between the effects of the two separators.

Experiments

In this section, we conduct a comprehensive experimental evaluation of PAGE. First, we systematically introduce the datasets used in the experiments. Second, we present SOTA methods for FU and GU employed for comparison. Finally, we describe the specific unlearning configurations and evaluation methodologies. Detailed experimental setup and analysis refer to (Ai et al. 2025). Overall, we aim to address the following research questions: **Q1**: Compared with the SOTA federated unlearning and graph unlearning methods, can PAGE achieve the best performance under multiple unlearn requests? **Q2**: How does PAGE achieve the dual advantages of maintaining model utility and forgetting integrity? **Q3**: Does PAGE demonstrates strong robustness across diverse unlearning scenarios?

Experimental Setup

All experiments are conducted on a system equipped with an NVIDIA A100 80GB PCIe GPU and an Intel(R) Xeon(R) Gold 6240 CPU @ 2.60GHz, with CUDA Version 12.4 enabled. The software environment is set up with Python 3.9.16 and PyTorch 1.13.1 to ensure optimal compatibility and performance for all algorithms.

Datasets. To comprehensively evaluate various unlearning methods, we collect real-world datasets of varying sizes from various domains (e.g., citation networks, co-author networks, etc.). From small-scale dataset Cora (Yang, Cohen, and Salakhutdinov 2016) to medium-scale dataset Photo (Shchur et al. 2018) to large-scale dataset ogbn-arxiv (Hu et al. 2020), we assign different numbers of clients according to the size of the dataset.

Baselines. We list Retrain and compare PAGE with 10 baseline methods. Most existing methods do not simultaneously support meta unlearning and client unlearning. Consequently, we divide the comparative experiments into two sets: (1) Client unlearning: FedEraser (Romandini et al. 2024), FUSED (Zhong et al. 2025), MoDe (Zhao et al. 2024), ReGEnUnlearn (Liu and Liu 2025). (2) Meta unlearning: GIF (Wu et al. 2023a), CEU (Wu et al. 2023b), D2DGN (Sinha, Mandal, and Kankanhalli 2024), MEGU (Li et al. 2024), FedLU (Zhu, Li, and Hu 2023), FedDM (Liu and Fang 2024).

Unlearning Settings. For the three unlearning modes in meta-unlearning, we randomly unlearn 10% of relevant graph data. Specifically, we randomly unlearn 10% of nodes for node unlearn request, 10% of edges for edge unlearn request and 10% of node features for feature unlearn request. For client unlearn, 20% of clients are randomly selected as unlearning clients, where all local data need to be unlearned.

Metrics. To evaluate the prediction performance of the model after unlearning, we use accuracy as the performance metric. For meta unlearning, we calculate the prediction accuracy of the local model on the local dataset. For client unlearn, we calculate the prediction accuracy of the global model on the local dataset. To rigorously validate unlearning effects, we employ membership inference attack (Olatunji, Nejdl, and Khosla 2021)(MIA), which can further determine whether the target instance has been effectively removed from the training set underlying the unlearned model.

	5 Clients		*10 Clients*		*15 Clients*		*20 Clients*	
	Cora	PubMed	CS	Photo	Tolokers	Minesweeper	Amazon-ratings	ogbn-arxiv
	Client Unlearning (Unlearn Ratio=0.2)							
Retrain	83.98 ± 1.52	83.90 ± 0.82	82.18 ± 0.43	78.41 ± 1.37	78.52 ± 1.51	79.88 ± 0.11	41.09 ± 0.48	57.04 ± 0.28
FedEraser	83.85 ± 2.37	78.52 ± 3.21	78.22 ± 1.46	54.66 ± 4.72	78.22 ± 1.61	79.81 ± 0.26	36.89 ± 0.40	28.63 ± 2.84
FUSED	49.43 ± 2.78	83.17 ± 0.40	72.68 ± 4.17	67.60 ± 5.34	78.47 ± 1.54	79.86 ± 0.10	39.08 ± 0.48	38.29 ± 0.43
MoDe	80.33 ± 4.04	83.20 ± 1.13	73.90 ± 3.42	42.47 ± 8.23	78.47 ± 1.54	79.86 ± 0.10	36.63 ± 4.19	28.31 ± 3.91
ReGEnUnlearn	83.83 ± 1.29	**83.74 ± 0.95**	78.27 ± 1.92	58.43 ± 9.78	**78.54 ± 1.51**	80.04 ± 0.21	40.20 ± 1.04	21.59 ± 5.30
Ours	**83.91 ± 1.29**	83.57 ± 0.54	**78.66 ± 1.06**	**69.73 ± 2.16**	78.52 ± 1.49	**80.06 ± 0.04**	**40.75 ± 0.58**	**50.13 ± 2.24**
	Meta Unlearning (Unlearn Ratio=0.1)							
Retrain	78.65 ± 0.04	83.91 ± 0.17	87.74 ± 0.09	88.62 ± 0.13	78.52 ± 0.18	79.90 ± 0.10	46.58 ± 0.06	69.37 ± 0.02
GIF	81.02 ± 0.38	84.21 ± 0.13	87.67 ± 0.05	80.44 ± 0.95	78.12 ± 0.08	79.69 ± 0.33	45.62 ± 0.13	67.37 ± 0.02
D2DGN	75.83 ± 0.61	83.23 ± 0.15	86.98 ± 0.33	79.69 ± 3.53	77.49 ± 0.26	76.14 ± 1.16	45.74 ± 0.20	55.19 ± 0.05
MEGU	77.96 ± 3.80	83.43 ± 0.05	86.94 ± 0.32	82.03 ± 3.03	78.34 ± 0.04	79.87 ± 0.00	42.63 ± 0.28	56.11 ± 0.13
FedLU	71.96 ± 0.43	73.14 ± 0.17	80.44 ± 0.16	80.21 ± 0.51	78.32 ± 0.04	79.87 ± 0.00	44.11 ± 0.09	62.75 ± 0.02
FedDM	80.11 ± 0.63	80.89 ± 0.20	80.67 ± 3.54	51.87 ± 2.57	76.82 ± 1.37	79.89 ± 0.08	32.09 ± 1.23	14.68 ± 3.03
Ours	**81.29 ± 0.17**	**84.39 ± 0.09**	**88.66 ± 0.07**	**88.84 ± 0.39**	**78.69 ± 0.19**	**79.92 ± 0.10**	**46.21 ± 0.13**	**68.19 ± 0.08**

Table 1: Performance Comparison. ACC ± STD(%) for node classification. The highest results are highlighted in **bold**.

Simulation Mode	Strategy	Physics		Computers	
		Edge-Level	Feature-Level	Edge-Level	Feature-Level
Metis	Retrain	92.99 ± 0.07	93.12 ± 0.16	87.09 ± 0.22	84.07 ± 4.47
	GIF	92.63 ± 0.03	93.01 ± 0.07	72.10 ± 4.79	71.96 ± 3.86
	D2DGN	88.12 ± 0.17	89.01 ± 0.28	71.51 ± 0.67	72.34 ± 1.63
	MEGU	92.59 ± 0.05	92.60 ± 0.03	73.01 ± 1.18	68.17 ± 4.91
	FedLU	91.15 ± 0.03	89.74 ± 0.78	79.89 ± 1.27	79.66 ± 0.19
	FedDM	77.28 ± 1.35	88.94 ± 1.90	45.88 ± 8.11	53.83 ± 0.59
	PAGE	**93.40 ± 0.03**	**93.52 ± 0.03**	**82.04 ± 0.43**	**82.42 ± 0.06**
Metis-Plus	Retrain	92.39 ± 0.01	92.70 ± 0.07	86.95 ± 0.23	86.07 ± 0.69
	GIF	90.37 ± 2.73	92.45 ± 0.18	74.18 ± 4.78	66.84 ± 6.14
	D2DGN	91.17 ± 0.18	84.95 ± 6.05	80.21 ± 0.53	80.26 ± 0.02
	MEGU	92.27 ± 0.06	92.08 ± 0.06	80.13 ± 0.01	77.18 ± 4.47
	FedLU	91.16 ± 0.06	89.84 ± 1.03	**83.26 ± 0.37**	84.74 ± 0.46
	FedDM	83.36 ± 2.89	91.81 ± 0.30	76.63 ± 1.13	72.49 ± 3.59
	PAGE	**92.66 ± 0.08**	**92.76 ± 0.06**	82.04 ± 0.33	**82.76 ± 0.59**

Table 2: Performance with edge and feature unlearning.

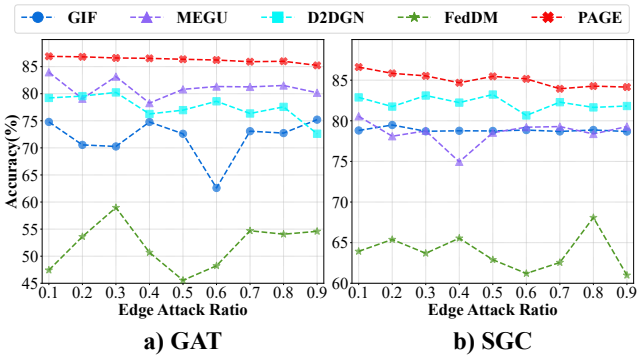

Figure 2: Performance on Photo under edge attack.

Q1: Performance Comparison

To answer Q1, we show the prediction performance comparison of each method under different unlearn request settings in Table 1. The results show that PAGE can outperform the baseline level on most datasets. On average, PAGE exhibits a remarkable average improvement of 5.08% over the SOTA approach under client unlearn request and 1.50% under meta unlearn request. It is noteworthy that on relatively small-scale datasets, the performance gap between PAGE and other methods is not statistically significant. However, on larger-scale datasets, PAGE demonstrates substantial performance advantages. For instance, under client unlearn requests, PAGE achieves merely a 2.13% improvement over SOTA methods on Photo, whereas it attains a 11.84% performance gain over SOTA methods on ogbn-arxiv. Additionally, for the other two granularity levels of unlearn requests (edge/feature unlearn) within meta-unlearn, we further evaluate PAGE's performance against comparative methods. As shown in Table 2, we compared the prediction performance of PAGE and other methods on Physics and Computers under simulation strategy metis-based community split (Metis) and metis-based label imbalance split (Metis-Plus). Except for some combinations, the prediction performance of PAGE is not as good as FedLU, but it is significantly ahead of other methods in other combinations.

To answer Q1 from the perspective of unlearning capability, Fig. 2 presents the unlearning capabilities of various methods under varying intensities of edge attack settings under meta unlearning. The edge-attack configuration involves injecting poisoned edges into each client's data before federated learning to degrade original model performance. After executing FGU algorithms, the models should regain their original performance, where the unlearning efficacy is quantified by the prediction accuracy (higher accuracy indicates stronger unlearning capability). This experiment employs the GAT (Veličković et al. 2018) and SGC (Wu et al. 2019) on Photo configurations to evaluate the prediction performance between methods under different edge-attack ratios. As demonstrated in Fig. 2, PAGE consistently maintains SOTA accuracy under multiple attack ratios, exhibiting minimal sensitivity to increasing attack intensity. For instance, on Photo, PAGE achieves 2.94% to 8.62% higher accuracy than comparative SOTA methods. These results substantiate that PAGE delivers superior unlearning capabilities.

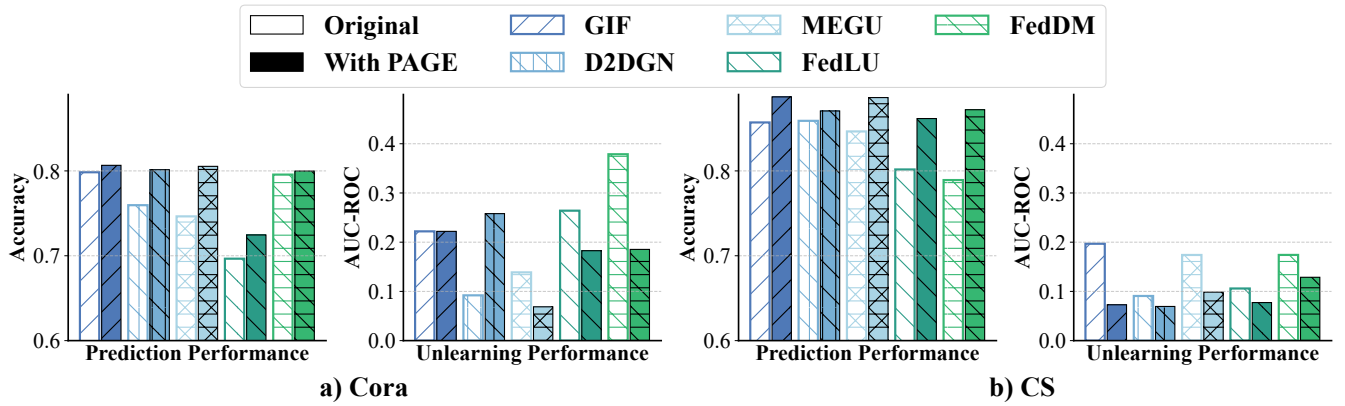

Figure 3: Augmentation study on GU methods. The left column represents original, the right represents combined with PAGE.

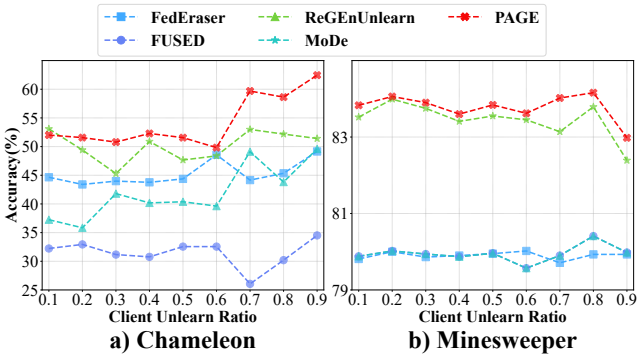

Figure 4: Performance under differnet client unlearn ratio.

Q2: Augmentation Study

In order to systematically verify the effectiveness of the components of our proposed method, we designed a series of exhaustive augmentation experiments to answer Q2. A key point of our method design is the dual role of the influenced unlearn component: in the client unlearn scenario, it operates independently as a core processing engine; while in the meta unlearn scenario, it acts as a plug-and-play enhancement plugin. The performance comparison in Q1 has verified the effectiveness of this component in client unlearn scenario, and here we verify its effectiveness as an enhancement plugin in meta unlearn scenario. As shown in Fig. 3, in the meta unlearn scenario, we evaluate the prediction performance and unlearning performance of the existing graph unlearning method before and after combining with the influenced unlearn component. As can be seen in the figure, the prediction performance and unlearning performance of the existing local unlearn method are significantly improved after combining with the influence module. Specifically, on Cora, the influenced unlearn component yields an average improvement of 2.82% in prediction performance and 4.64% in unlearning performance compared to the base method. On CS, it further enhances prediction performance by 4.49% and unlearning performance by 7.22%.

Q3: Robustness Analysis

To answer Q3, we designed experiments that evaluate model performance under varying unlearn intensities. Using GraphSAGE as the backbone under client unlearn with louvain simulation, we measured prediction accuracy across different unlearn ratios on Chameleon and Minesweeper.

As shown in Fig. 4, PAGE maintains superior performance over counterparts despite increasing unlearn intensity. On dataset Chameleon, while slightly underperforming RFPS at low unlearn intensities, PAGE demonstrates minimal fluctuation and remains stable as intensity escalates. Notably, at high forgetting intensities, PAGE exhibits counterintuitive performance gains—increasing accuracy over baselines and widening the performance gap. From the above experiments, it can be seen that compared with other methods, PAGE has stronger robustness.

Furthermore, Fig. 2 reveals that as the edge-attack ratio progressively increases, PAGE persistently maintains prediction stability without exhibiting significant performance fluctuations observed in other methods. This robustness against adversarial edge perturbations substantiates PAGE's exceptional resilience.

Conclusion

This paper critically analyzes federated and graph unlearning methods, revealing incomplete unlearning support and residual knowledge permeation as fundamental limitations. To address these gaps, we introduce the first multi-scenario federated graph unlearning framework. Our approach pioneers prototype matching while integrating negative knowledge distillation to systematically eliminate knowledge permeation. Core innovations include: (1) Utilizing prototype vectors as knowledge carriers for precise local unlearning; (2) Extracting unlearned knowledge via adversarial graph samples; (3) Directing cross-client influence unlearning to eradicate permeation impacts without privacy compromise. Future research must prioritize integrated frameworks harmonizing local and influence unlearning for comprehensive permeation mitigation.

Acknowledgments

This work was supported by the NSFC Grants U2241211, U24A20255 and 62427808. Rong-Hua Li is the corresponding author of this paper.

References

Ai, Y.; Li, X.; Chao, J.; Fan, B.; Wu, Z.; Zhu, Y.; Li, R.-H.; and Wang, G. 2025. Federated Graph Unlearning. arXiv:2508.02485.

Cai, X.; Huang, C.; Xia, L.; and Ren, X. 2023. LightGCL: Simple Yet Effective Graph Contrastive Learning for Recommendation. In *International Conference on Learning Representations, ICLR*.

Chang, H.; and Shokri, R. 2023. Bias propagation in federated learning. *arXiv preprint arXiv:2309.02160*.

Dong, Y.; Zhang, B.; Lei, Z.; Zou, N.; and Li, J. 2024. IDEA: A Flexible Framework of Certified Unlearning for Graph Neural Networks. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 621–630. Association for Computing Machinery. ISBN 9798400704901.

He, C.; Balasubramanian, K.; Ceyani, E.; Yang, C.; Xie, H.; Sun, L.; He, L.; Yang, L.; Yu, P. S.; Rong, Y.; et al. 2021. Fedgraphnn: A federated learning system and benchmark for graph neural networks. *arXiv preprint arXiv:2104.07145*.

Hu, W.; Fey, M.; Zitnik, M.; Dong, Y.; Ren, H.; Liu, B.; Catasta, M.; and Leskovec, J. 2020. Open graph benchmark: Datasets for machine learning on graphs. *Advances in Neural Information Processing Systems, NeurIPS*.

Li, X.; Zhao, Y.; Wu, Z.; Zhang, W.; Li, R.-H.; and Wang, G. 2024. Towards Effective and General Graph Unlearning via Mutual Evolution. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(12): 13682–13690.

Liu, B.; and Fang, Y. 2024. Federated knowledge graph unlearning via diffusion model. *arXiv preprint arXiv:2403.08554*.

Liu, F.; and Liu, H. 2025. Subgraph Federated Unlearning. In *THE WEB CONFERENCE 2025*.

Olatunji, I. E.; Nejdl, W.; and Khosla, M. 2021. Membership Inference Attack on Graph Neural Networks. In *2021 Third IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*, 11–20.

Pardau, S. L. 2018. The california consumer privacy act: Towards a european-style privacy regime in the united states. *J. Tech. L. & Pol'y*, 23: 68.

Qu, Z.; Yao, T.; Liu, X.; and Wang, G. 2023. A Graph Convolutional Network Based on Univariate Neurodegeneration Biomarker for Alzheimer's Disease Diagnosis. *IEEE Journal of Translational Engineering in Health and Medicine*.

Regulation, G. D. P. 2018. General data protection regulation (GDPR). ntersoft Consulting. *Obtenido de https://www. epsu. org/sites/¿ default/files/article/-files/GDPR_FINAL_EPSU. pdf*.

Romandini, N.; Mora, A.; Mazzocca, C.; Montanari, R.; and Bellavista, P. 2024. Federated unlearning: A survey on methods, design guidelines, and evaluation metrics. *IEEE Transactions on Neural Networks and Learning Systems*.

Shchur, O.; Mumme, M.; Bojchevski, A.; and Günnemann, S. 2018. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868*.

Sinha, Y.; Mandal, M.; and Kankanhalli, M. 2024. Distill to Delete: Unlearning in Graph Networks with Knowledge Distillation. arXiv:2309.16173.

Tolpegin, V.; Truex, S.; Gursoy, M. E.; and Liu, L. 2020. Data poisoning attacks against federated learning systems. In *European symposium on research in computer security*, 480–501. Springer.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2018. Graph attention networks. In *International Conference on Learning Representations, ICLR*.

Wang, Z.; Chen, T.; Ren, J.; Yu, W.; Cheng, H.; and Lin, L. 2018. Deep Reasoning with Knowledge Graph for Social Relationship Understanding. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, 1021–1028. International Joint Conferences on Artificial Intelligence Organization.

Wu, F.; Souza, A.; Zhang, T.; Fifty, C.; Yu, T.; and Weinberger, K. 2019. Simplifying graph convolutional networks. In *International Conference on Machine Learning, ICML*.

Wu, J.; Yang, Y.; Qian, Y.; Sui, Y.; Wang, X.; and He, X. 2023a. GIF: A General Graph Unlearning Strategy via Influence Function. In *Proceedings of the ACM Web Conference, WWW*.

Wu, K.; Shen, J.; Ning, Y.; Wang, T.; and Wang, W. H. 2023b. Certified Edge Unlearning for Graph Neural Networks. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2606–2617. Association for Computing Machinery. ISBN 9798400701030.

Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; and Yu, P. S. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1): 4–24.

Yang, Y.; Han, X.; Chai, Y.; Ebrahimi, R.; Behnia, R.; and Padmanabhan, B. 2024. From Machine Learning to Machine Unlearning: Complying with GDPR's Right to be Forgotten while Maintaining Business Value of Predictive Models. *arXiv preprint arXiv:2411.17126*.

Yang, Z.; Cohen, W. W.; and Salakhutdinov, R. 2016. Revisiting Semi-Supervised Learning with Graph Embeddings. In *International Conference on Machine Learning, ICML*.

Zhao, Y.; Wang, P.; Qi, H.; Huang, J.; Wei, Z.; and Zhang, Q. 2024. Federated Unlearning With Momentum Degradation. *IEEE Internet of Things Journal*, 11(5): 8860–8870.

Zhong, Z.; Bao, W.; Wang, J.; Zhang, S.; Zhou, J.; Lyu, L.; and Lim, W. Y. B. 2025. Unlearning through knowledge overwriting: Reversible federated unlearning via selective sparse adapter. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 30661–30670.

Zhu, X.; Li, G.; and Hu, W. 2023. Heterogeneous federated knowledge graph embedding learning and unlearning. In *Proceedings of the ACM web conference 2023*, 2444–2454.