FedUP: One-Shot Federated Unlearning via Centroid-Guided Plug-in Filters

Feihong Nan[1] , **Zhengyi Zhong**[1] , **Pan Wang**[1] , **Weidong Bao**[1] , **Xiongtao Zhang**[1] , **Quan Wen**[1] and **Ji Wang**[1*]

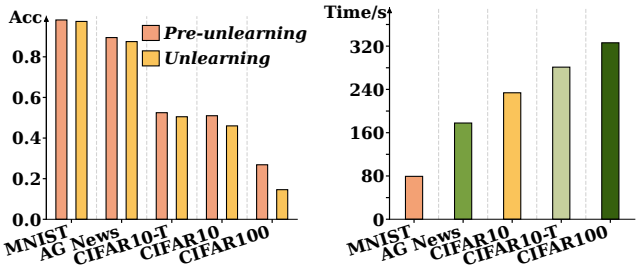
[1]National Key Laboratory of Big Data and Decision, National University of Defense Technology, China.
feihongnan178@outlook.com, {zhongzhengyi20, wangpan19, wdbao, zhangxiongtao14, wangji}@nudt.edu.cn, weixingw1@sina.com

Abstract

Federated unlearning (FU) is critical for complying with legal mandates like the right to be forgotten in decentralized systems, yet current methods face a persistent dilemma between *non-target knowledge loss* and *high request latency*. To resolve these issues, we propose FedUP, a one-shot federated unlearning framework utilizing lightweight pluggable filters that act as a "knowledge funnel" to screen out target data while preserving original model performance. By freezing original model parameters and training filters at the server side using differentially private (DP)-protected class centroid samples, FedUP bypasses the need for multi-round client-server communication and complex retraining, reducing unlearning latency from minutes to mere seconds. Additionally, the framework's pluggable architecture ensures inherent reversibility, enabling the seamless restoration of forgotten knowledge by simply removing the filters. Extensive experiments on diverse image and text tasks demonstrate that FedUP effectively reduces non-target knowledge loss and achieves superior unlearning precision and efficiency across various scenarios. Code is available at: https://github.com/suows/FedUP-code.

1 Introduction

Background. As a distributed machine learning paradigm, federated learning (FL) [McMahan *et al.*, 2016; Truong *et al.*, 2021; Zhong *et al.*, 2022; Qi *et al.*, 2024; Zhong *et al.*, 2025a; Fu *et al.*, 2025b; Jiang *et al.*, 2026] has gained significant attention in privacy-sensitive scenarios recently because it eliminates the need for centralizing raw data during training. However, during the training process, the global model internalizes client information into its parameters through multiple rounds of parameter aggregation, which exposes novel privacy risks when the model is confronted with legal mandates such as the right to be forgotten under GDPR [de la Torre, 2018]. To address this, researchers have focused on Federated Unlearning (FU), aiming to remove specific knowledge without retraining the global model from scratch. Existing FU methods generally follow two main technical paradigms: server-side FU methods [Wu *et al.*, 2022; Huynh *et al.*, 2025; Pan *et al.*, 2025], which implement approximate unlearning [Yang *et al.*, 2025] on the global model at the server side; and client-side FU methods [Wang *et al.*, 2023b; Liu *et al.*, 2022; Zhu *et al.*, 2023; Zhong *et al.*, 2025b], which achieve exact unlearning [Kuo *et al.*, 2025] through iterative client-side model retraining procedures.



(a) Non-target knowledge loss of server-side FU method. (b) Unlearning request latency of client-side FU method.

Figure 1: FedEraser, a representative server-side method, shows noticeable non-target knowledge loss after unlearning across multiple tasks. Client-side methods like FUSED experience significantly longer convergence times as task difficulty increases, with responses up to 5 minutes indicating severe unlearning request latency.

Existing Challenges. While current Federated Unlearning (FU) methods facilitate knowledge removal to various extents, challenges regarding non-target knowledge loss and unlearning request latency still remain. As depicted in Figure 1, server-side FU methods lack precise control over the unlearning scope, frequently incurring ***non-target knowledge loss***, where the model performance unrelated to the unlearning request is inevitably impaired during knowledge removal. Conversely, although client-side FU methods achieve exact unlearning via retraining-based approaches, they are heavily constrained by multi-round training and communication, leading to ***high request latency***. To date, it is hard to find a solution that simultaneously ensures rapid response and prevents non-target knowledge loss. Moreover, both server-side and client-side FU paradigms typically rely on direct modification of original model parameters, which leads to ***the irreversibility of unlearning***. Once the unlearning process is

*Corresponding Author

finalized, restoring previously removed knowledge necessitates further parameter tuning or retraining, thereby imposing complexity and additional overhead on practical deployment.

Proposed Solution. To this end, we propose FedUP, a one-shot federated unlearning framework based on lightweight pluggable filters. In terms of mitigating non-target knowledge loss, the framework avoids performing direct, large-scale parameter updates on the global model; instead, it implements unlearning by introducing independent pluggable filters while keeping original model parameters frozen (shown in Figure 2). These filters serve as a "knowledge funnel" that screens out target knowledge and permits only non-target knowledge to pass, thereby reducing the interference of unlearning on non-target knowledge. To lower unlearning request latency, FedUP only needs to perform a few rounds of fine-tuning on the lightweight filters at the server side using differential privacy (DP)-protected class centroid samples to complete the unlearning task. This bypasses multi-round retraining and frequent communication with clients, significantly accelerating the response speed. Moreover, since the filters are pluggable, the framework inherently supports reversibility. As depicted in Figure 2, when restoration of forgotten knowledge is required, it can be achieved simply by removing the filters. Overall, FedUP provides a solution that balances precision, efficiency, and recoverability.

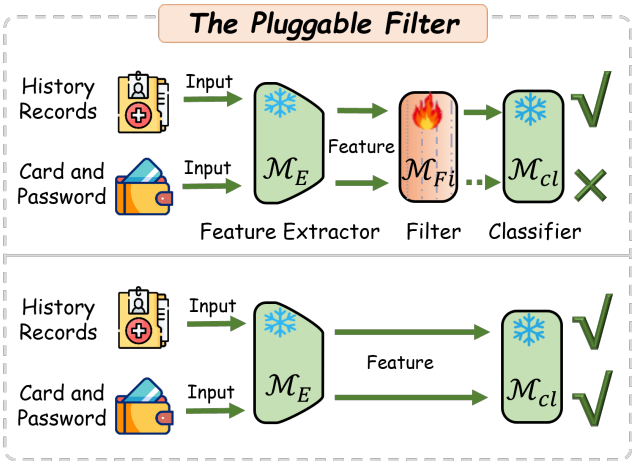


Figure 2: Lightweight plug-in filters.

Contributions. The main contributions are as follows:

- We design FedUP, a one-shot FU framework utilizing DP-protected class centroids, mitigating non-target knowledge loss and reducing unlearning request latency from minutes to seconds.
- We propose a reversible unlearning mechanism via lightweight pluggable filters without altering original model parameters while ensuring rapid transitions between unlearned and pre-unlearning states.
- We conduct differential privacy analysis showing that appropriate noise protects class centroid samples while minimally impacting data utility. Extensive experiments on image and text tasks confirm FedUP's effectiveness across various scenarios.

2 Related Work

Machine Unlearning. Simply removing training data from storage fails to purge its influence on deployed models. Machine Unlearning (MU) [Ma *et al.*, 2022] is thus introduced to erase such learned knowledge. Existing methods are categorized as either exact [Cao *et al.*, 2018] or approximate unlearning [Golatkar *et al.*, 2020]. Exact unlearning mandates that the post-forgetting model be statistically indistinguishable from one retrained from scratch without the deleted data. It retains the guarantees of complete retraining but lowers its cost via algorithmic shortcuts. For classical models, the training process can be expressed in an invertible additive form, enabling point removal by subtracting its closed-form term [Cao *et al.*, 2018]. For complex architectures, exact unlearning employs data sharding [Bourtoule *et al.*, 2021], localized retraining [Chen *et al.*, 2022], or intermediate checkpointing [Wang *et al.*, 2023a] to reduce computation while preserving retraining-level guarantees. Approximate unlearning, in contrast, substitutes full retraining with lightweight fine-tuning, permitting a bounded, residual influence from the deleted data. Related work is typically grouped into data-driven and model-driven approaches [Nguyen *et al.*, 2025]. Data-driven methods involve relabeling retained samples [Graves *et al.*, 2021] or partitioning data shards [Gupta *et al.*, 2021] before fine-tuning. Model-driven approaches adjust parameters directly via influence functions [Guo *et al.*, 2019], fisher-based regularization [Golatkar *et al.*, 2020], or knowledge distillation [Kurmanji *et al.*, 2023], countering the gradient contributions of the data to be erased.

Federated Unlearning. Though federated learning maintains client privacy through local retention, knowledge from distributed datasets persists in the aggregated global model. This requires integrating MU techniques into FL, named federated unlearning (FU) [Wang *et al.*, 2022]. Based on the execution location, FU methods can be categorized into server-side and client-side methods [Liu *et al.*, 2024; Li *et al.*, 2025]. Server-side methods are intrinsically based on approximate unlearning to expunge client contributions without client involvement. FedEraser [Liu *et al.*, 2021] calibrates update trajectories but still requires auxiliary retraining. Conversely, Wu et al. [Wu *et al.*, 2022] bypass client-side computation by subtracting historical updates and employing knowledge distillation. To optimize efficiency, Huynh et al. [Huynh *et al.*, 2025] utilize selective retention of influential updates to reduce memory overhead, while Pan et al. [Pan *et al.*, 2025] resolve parameter conflicts via Orthogonal Steepest Descent to accelerate the unlearning process. Client-side FU methods perform unlearning locally, striving to achieve exact unlearning while balancing computational efficiency with global model utility. Mora et al. [Mora *et al.*, 2024] propose FedUNRAN, utilizing local random label perturbations to disperse target gradients and attenuate their influence without server-side intervention. Wang et al. [Wang *et al.*, 2023b] employ variational Bayesian inference for parameter self-sharing to erase target data while preserving performance. To accelerate unlearning, Liu et al. [Liu *et al.*, 2022] approximate the Hessian via a diagonal empirical Fisher Information Matrix for quasi-Newton optimization, while Zhu et al. [Zhu

et al., 2023] combine inverse perturbation with passive decay, propagating updates via knowledge distillation. Furthermore, Deng et al. [Deng *et al.*, 2024] introduce model contrastive unlearning (MCU) to regularize the feature space. Zhong et al. [Zhong *et al.*, 2025b] implement reversible unlearning through local fine-tuning.

In summary, existing unlearning approaches typically involve trade-offs between unlearning precision and computational efficiency, and often suffer from limited generalization. Moreover, reversibility is rarely considered in current studies. Achieving a unified balance among unlearning precision, efficiency, and reversibility therefore remains an open challenge.

3 Methodology

3.1 Problem Description

In the FL framework, a set of clients denoted as $\mathcal{C} = \{C_1, C_2, \ldots, C_N\}$ collaboratively train a global model $\mathcal{M}_G$ without sharing local data. Each client C_n trains a local model $\mathcal{M}_n$ using its local dataset $\mathcal{D}_n$ and uploads the model parameters to a server. The server aggregates these local models via weighted averaging based on dataset sizes to obtain a global model $\mathcal{M}_G$. This process iterates over W global rounds, where each global round comprises e local training epochs on the clients with a local learning rate l_c. $\mathcal{M}_G$ is structurally decomposed into a feature extractor $\mathcal{M}_E$ and a classifier $\mathcal{M}_{cl}$. After aggregation, $\mathcal{M}_G$ is distributed back to all clients, where the feature extractor $\mathcal{M}_E$ is leveraged to extract features from local data. Additionally, a pluggable filter $\mathcal{M}_{Fi}$ is constructed to implement the filtering of knowledge that needs to be forgotten. The overall dataset is denoted as $\mathcal{D} = \{\mathcal{D}_n\}_{n=1}^{N}$, where $\mathcal{D}_n$ represents the local dataset of client C_n. The total data volume across the federation is $\|\mathcal{D}\| = \sum_{n=1}^{N} \|\mathcal{D}_n\|$. To facilitate data management, particularly for future unlearning operations, each local dataset $\mathcal{D}_n$ is partitioned into two disjoint subsets: a retained subset $\mathcal{D}_n^R$ for model training, and an unlearning subset $\mathcal{D}_n^U$ designated to be removed in compliance with privacy or regulatory constraints. Let $k \in \{1, 2, \ldots, K\}$ index the data categories, where K is the total number of classes.

In the FL phase, the training objective is defined as:

$$\min_{\theta_{\mathcal{M}_G}} F(\theta_{\mathcal{M}_G}) = \sum_{n=1}^{N} \frac{|\mathcal{D}_n|}{|\mathcal{D}|} \sum_{(x_i^k, y_i^k) \in \mathcal{D}_n} \mathcal{L}\big(f(x_i^k; \theta_{\mathcal{M}_G}), y_i^k\big), \tag{1}$$

where $\mathcal{L}$ denotes the loss function. During the FU phase, the objective is to maximize the loss on the unlearning set, minimize the loss on the retained set, and keep the training cost low, described as:

$$\min_{\theta_{\mathcal{M}'_G}} F^R(\theta_{\mathcal{M}'_G}) = \sum_{n=1}^{N} \frac{|\mathcal{D}_n^R|}{|\mathcal{D}^R|} \sum_{(x_i^k, y_i^k) \in \mathcal{D}_n^R} \mathcal{L}\big(f(x_i^k; \theta_{\mathcal{M}'_G}), y_i^k\big), \tag{2}$$

$$\max_{\theta_{\mathcal{M}'_G}} F^U(\theta_{\mathcal{M}'_G}) = \sum_{n=1}^{N} \frac{|\mathcal{D}_n^U|}{|\mathcal{D}^U|} \sum_{(x_i^k, y_i^k) \in \mathcal{D}_n^U} \mathcal{L}\big(f(x_i^k; \theta_{\mathcal{M}'_G}), y_i^k\big), \tag{3}$$

$$\min F^T(\theta_{\mathcal{M}'_G}) = \sum_{w=1}^{W} \mathrm{Comm}^{(w)}(\mathcal{M}_G, \mathcal{C}), \tag{4}$$

where Comm represents the communication resource between the server and the clients $\mathcal{C}$ during each round.

3.2 Method Overview

As shown in Figure 3, our method comprises four phases: federated learning, generation of class centroid samples, one-shot federated unlearning, and inference, with each stage's main procedures and key formulas shown in the diagram.

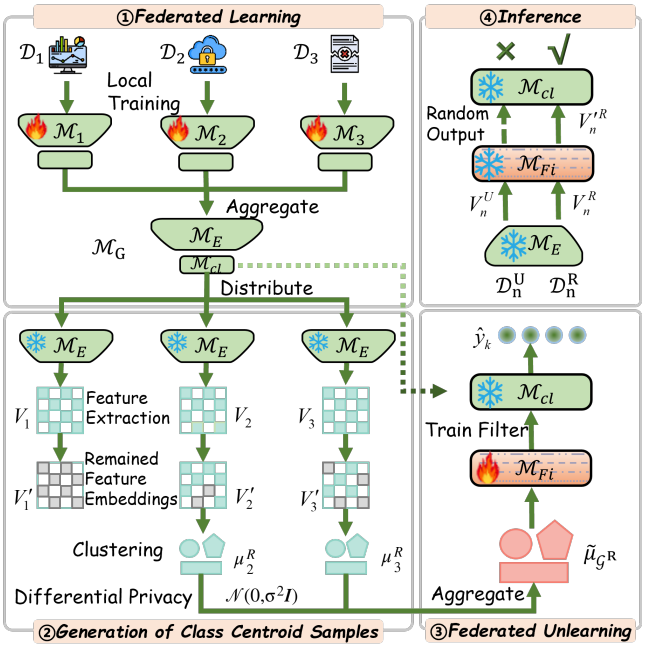


Figure 3: FedUP follows a structured workflow: it begins with federated learning to obtain a global model. Upon request, clients produce differentially-private class centroid samples from retained data and upload them for server aggregation. The server then fine-tunes a pluggable filter, blocking forgotten knowledge without modifying the base model.

Feature Extraction

Each client C_n performs e rounds of training on its local dataset $\mathcal{D}_n$, resulting in an updated local model $\mathcal{M}_n^{w,e}$. The local training process can be expressed as:

$$\theta_{\mathcal{M}_n}^{w,e} = \theta_{\mathcal{M}_n}^{w,e-1} - l_c \cdot \nabla_\theta \mathcal{L}(\mathcal{M}_n^{w,e-1}, \mathcal{D}_n), \tag{5}$$

where $\mathcal{L}$ represents the local loss function and $\nabla_\theta \mathcal{L}$ denotes its gradient with respect to the model parameters.Subsequently, the server collects the local models $\mathcal{M}_n^{w,e}$ from selected clients C_n and updates the global model through a weighted average:

$$\theta_{\mathcal{M}_G}^{w+1} = \sum_{n=1}^{N} \frac{|\mathcal{D}_n|}{|\mathcal{D}|} \theta_{\mathcal{M}_n}^{w,e}. \tag{6}$$

The trained global model $\mathcal{M}_G$ is regarded as a combination of a feature extractor $\mathcal{M}_E$ and a classifier $\mathcal{M}_{cl}$. After freezing this composite model, it is deployed to the clients. The features of all data $\mathcal{D}_n$ from each client C_n are extracted by the feature extractor $\mathcal{M}_E$, which can be represented as follows:

$$V_n^k = \left\{ \mathcal{M}_E(x_i^k) \mid x_i^k \in \mathcal{D}_n^k \right\}, \tag{7}$$

where V_n^k denotes the feature embedding of the data belonging to class k on client C_n.

Generation of Class Centroid Samples

Upon receiving an unlearning request, client C_n removes the features associated with the unlearning data. The remaining feature embeddings belonging to class k are denoted as:

$$V_n^{R,k} = V_n^k \setminus \left\{ \mathcal{M}_E(x_i) \mid x_i \in \mathcal{D}_n^{U,k} \right\}. \tag{8}$$

To reduce communication cost, the retained features are compressed via KMeans clustering with

$$K_{n,k} = \lceil \rho \, |V_n^{R,k}| \rceil, \tag{9}$$

the number of clusters is set proportionally according to the scenario by ρ, yielding a compact set of centroids:

$$\mu_n^{R,k} = \mathrm{KMeans}(V_n^{R,k}, K_{n,k}). \tag{10}$$

For privacy preservation, add noise to each centroid:

$$\tilde{\mu}_n^{R,k} = \left\{ \mu + \mathbf{z} \;\middle|\; \mu \in \mu_n^{R,k},\ \mathbf{z} \sim \mathcal{N}(0, \sigma^2 I) \right\}. \tag{11}$$

Here, $\mathcal{N}(0, \sigma^2 \mathbf{I})$ represents a multivariate Gaussian distribution. These mechanisms ensure that the release of $\tilde{\mu}_n^{R,k}$ does not reveal excessive information about any individual data point. Client C_n then uploads all perturbed class centroids $\tilde{\mu}_n^{R,k}$ to the server. Using the local class centroids $\tilde{\mu}_n^{R,k}$, the server aggregates them to generate the global class centroids $\tilde{\mu}_{\mathcal{G}}^{R,k}$. The global class centroid is denoted as:

$$\tilde{\mu}_{\mathcal{G}}^{R,k} = \left[\tilde{\mu}_1^{R,k}; \tilde{\mu}_2^{R,k}; \ldots; \tilde{\mu}_N^{R,k} \right]. \tag{12}$$

One-shot Federated Unlearning

The objective of the FU phase is to train the filter $\mathcal{M}_{Fi}$ such that it effectively blocks the flow of unlearning knowledge while allowing the retained learning knowledge to pass. The specific steps and formulas are as follows.

A filter $\mathcal{M}_{Fi}$ is inserted into the global model $\mathcal{M}_G = \mathcal{M}_E \oplus \mathcal{M}_{cl}$, yielding a new model structure:

$$\mathcal{M}'_G = \mathcal{M}_E \oplus \mathcal{M}_{Fi} \oplus \mathcal{M}_{cl}. \tag{13}$$

Subsequently, the parameters of the global model $\mathcal{M}_G$ including both $\mathcal{M}_E$ and $\mathcal{M}_{cl}$ are frozen:

$$\theta_{\mathcal{M}_\mathcal{G}} = \theta^*_{\mathcal{M}_\mathcal{G}}. \tag{14}$$

The filter $\mathcal{M}_{Fi}$ is trained using the global class centroids $\tilde{\mu}_{\mathcal{G}}^{R,k}$. The filtered class centroid prediction is defined as:

$$\{\hat{y}_k\} = \{\mathcal{M}_{\mathrm{cl}}(\mathcal{M}_{\mathrm{Fi}}(\tilde{\mu})) \mid \tilde{\mu} \in \tilde{\mu}_{\mathcal{G}}^{R,k}, k \in \mathcal{K}_R\}. \tag{15}$$

We propose a composite loss function for the filter consisting of cross-entropy and reconstruction losses. It preserves discriminative capability on non-target knowledge while enforcing structural consistency in the feature space, thereby improving the stability of the unlearning process without compromising knowledge selectivity. The cross-entropy loss, measuring the difference between predicted class probabilities and true labels, is defined as:

$$\mathcal{L}_{\mathrm{CE}} = -\sum_k y_k \log(\hat{y}_k), \tag{16}$$

where y_k represents the true label while $\hat{y}_k$ denotes the corresponding predicted probability for class k. The reconstruction loss, which measures how well the filter reconstructs its input, is defined as:

$$\mathcal{L}_{\mathrm{RE}} = \frac{1}{d} \sum_{i=1}^{d} |\tilde{\mu} - \mathcal{M}_{Fi}(\tilde{\mu})|^2, \tag{17}$$

where d is the input and output dimensionality of the filter. The total loss function used to train the filter $\mathcal{M}_{Fi}$ is a weighted sum of these two losses:

$$\mathcal{L}_{\mathrm{total}} = \alpha \mathcal{L}_{\mathrm{CE}} + (1 - \alpha) \mathcal{L}_{\mathrm{RE}}, \tag{18}$$

where α is a weighting factor that balances the importance of the cross-entropy loss and the reconstruction loss. Train the filter for W_a rounds with a learning rate l_a. The update process is as follows:

$$\theta^w_{\mathcal{M}_{Fi}} = \theta^{w-1}_{\mathcal{M}_{Fi}} - l_a \cdot \nabla_\theta \mathcal{L}_{\mathrm{total}} \left(\theta^{w-1}_{\mathcal{M}_{Fi}}; \tilde{\mu}_{\mathcal{G}}^{R,k} \right) \quad w = 1, \ldots, W_a. \tag{19}$$

When a client requests the restoration of forgotten knowledge, the filter $\mathcal{M}_{Fi}$ can be removed, thereby reverting to the original global model $\mathcal{M}_G$:

$$\mathcal{M}_G = \mathcal{M}_E \oplus \mathcal{M}_{cl}. \tag{20}$$

Differential Privacy Guarantee

Our framework injects Gaussian noise for privacy guarantee during client-side centroid uploads, satisfying (ε, δ)-DP. The key is to calibrate the noise scale σ according to the desired privacy budget and the ℓ_2-sensitivity, which is bounded by:

$$\Delta_2 f_i = \max_{p,q} \frac{\|d_p^{(i)} - d_q^{(i)}\|_2}{n_i}. \tag{21}$$

Noise / *Dataset*	σ = 0.001	0.005	0.01	0.05	0.1
MNIST	**11.09**	2.22	1.11	0.22	0.11
CIFAR-10	**16.73**	3.35	1.67	0.33	0.17
AG News	**12.27**	2.45	1.23	0.25	0.12
CIFAR-100	141.40	28.28	**14.14**	2.83	1.41

Table 1: Average privacy budget for different datasets.

We set $\sigma \geq \frac{\sqrt{2\ln(1.25/\delta_i)} \cdot \Delta_2 f_i}{\varepsilon_i}$ with $\delta_i = 1/n_i$. Empirical ε_i values (Table 1) confirm moderate privacy budgets $\varepsilon \approx 10$ [Wei *et al.*, 2020] for all datasets under specified σ. We calculate our differentially private guarantee as:

Algorithm 1: FedUP

Input: Number of global rounds W, local rounds e, local learning rate l_c, adapter learning rate l_a, clients $\mathcal{C} = \{C_1, C_2, \ldots, C_N\}$, dataset $\mathcal{D} = \{\mathcal{D}_n\}_{n=1}^{N}$
Output: Filter $\mathcal{M}_{Fi}$
/ * Feature extraction * /
$\mathcal{M}_G = \mathcal{M}_E \oplus \mathcal{M}_{cl}$
for *global round* $w = 1$ ***to*** W **do**
  Server sends $\mathcal{M}_G^w$ to all clients C_n
  for *each client* C_n **do**
    C_n performs e local training rounds:
    $\theta_{\mathcal{M}_n}^{w,e} = \theta_{\mathcal{M}_n}^{w} - l_c \cdot \nabla_\theta \mathcal{L}(\mathcal{M}_n, \mathcal{D}_n^k)$
  $\theta_{\mathcal{M}_G}^{w+1} = \sum_{n=1}^{N} \frac{|\mathcal{D}_n|}{|\mathcal{D}|} \theta_{\mathcal{M}_n}^{w,e}$
/ * Generation of class centroid samples * /
for *each client* C_n **do**
  $V_n^k = \mathcal{M}_E(x_i^k), x_i^k \in \mathcal{D}_n$
  $V'^k_n = V_n^k \setminus \{\mathcal{M}_E(x_i^k) | x_i^k \in \mathcal{D}_n^U\}$
  $\mu_n^{R,k} = \mathrm{KMeans}(V_n^{R,k}, K_{n,k})$
  Generate $\tilde{\mu}_n^{R,k}$ via Eq. (11)
/ * One-shot federated unlearning * /
$\mathcal{M}'_G = \mathcal{M}_E \oplus \mathcal{M}_{Fi} \oplus \mathcal{M}_{cl}$
Freeze parameters of $\mathcal{M}_\mathcal{G}$: $\theta_{\mathcal{M}_\mathcal{G}} = \theta^*_{\mathcal{M}_\mathcal{G}}$
Train filter $\mathcal{M}_{Fi}$ using $\tilde{\mu}_\mathcal{G}^{R,k}$ via Eq. (18) and Eq. (19)
/ * Restoration * /
Remove filter: $\mathcal{M}_{Fi}$: $\mathcal{M}_G = \mathcal{M}_E \oplus \mathcal{M}_{cl}$

$$\varepsilon_i = \frac{\sqrt{2\ln(1.25n_i)} \cdot \Delta_2 f_i}{\sigma}. \tag{22}$$

The final choice of the optimal σ is determined via $\sigma = q \cdot s$, with the corresponding values of q and s established accordingly. Detailed analysis and supporting experiments are relegated to Section 2.1 and 2.2 of the supplementary material.

3.3 Algorithm

The algorithm, as illustrated in Algorithm 1, consists of three stages: feature extraction, generation of class centroid samples $\mu_n^{R,k}$ and one-shot federated unlearning. Upon receiving an unlearning request, each client first removes the corresponding data points from its local feature set and computes class centroid samples using the retained data. These centroids are then protected by a differential privacy mechanism and uploaded to the server. The server aggregates the uploaded centroids from all clients to obtain global retained class centroids as $\tilde{\mu}_\mathcal{G}^{R,k}$. Then the server freezes the parameters of the original global model $\mathcal{M}_\mathcal{G}$ and randomly initializes a lightweight, pluggable filter $\mathcal{M}_{Fi}$, which is trained solely using the global differentially private centroids. By jointly optimizing a cross-entropy loss and a reconstruction loss $\mathcal{L}_{\text{total}}$, the filter blocks the propagation of forgotten knowledge while preserving the discriminative capability of retained knowledge. Finally, when restoration of forgotten knowledge is required, the filter can be removed to revert to the original global model structure, enabling efficient and reversible federated unlearning.

4 Experiment

4.1 Experimental Setup

We conducted experiments on diverse datasets, including MNIST[LeCun *et al.*, 2002], CIFAR-10, CIFAR-100 [Krizhevsky *et al.*, 2009], and AG News[Zhang *et al.*, 2015]. We partitioned the dataset using the Dirichlet distribution [Li *et al.*, 2022] with a concentration parameter of 0.5. These experiments use various network architectures, including LeNet-5, ResNet-18 [He *et al.*, 2016], ResNet-34, Transformer [Vaswani *et al.*, 2017] and TinyBert[Jiao *et al.*, 2020], covering three unlearning scenarios: client unlearning, class unlearning and sample unlearning. We evaluated five baselines: EraseClient [Halimi *et al.*, 2022], Federaser [Liu *et al.*, 2021], Exact-Fun [Xiong *et al.*, 2023], Retrain and FUSED [Zhong *et al.*, 2025b]. The experimental framework is implemented using PyTorch 2.3.1 and CUDA 12.1. For hardware acceleration, an NVIDIA RTX 3080 Ti GPU is utilized. We employ SGD and Adam optimizers. The balancing factor α for the loss of filters is set to 0.5. During the generation of sample centroids via clustering, we set different sampling ratios ρ to accommodate various scenarios. In the client, class, and sample scenarios, the sampling ratios are 0.8, 0.1, and 0.5, respectively.

Metric	*Description*
R-A(%)↑	Accuracy of retained knowledge.
F-A(%)↓	Accuracy of unlearning knowledge.
0A(%)↑	Accuracy of class 0.
PS(%)↑	Prediction precision of class 0.
MIA(%)↓	Post-unlearning privacy risk quantified via membership inference attacks.
Time(s)↓	Time required to achieve the desired effect.
Comm(MB)↓	Communication resource consumption between client and server.

Table 2: Evaluation metrics of FU.

Evaluation Metrics. As shown in Table 2, we evaluate our proposed method and the baselines using multiple metrics.

4.2 Experimental Results

Main Results. In the client unlearning setting, Byzantine attacks are introduced, where label flipping is applied to construct inverse feature prototypes [Fu *et al.*, 2025a; Qi *et al.*, 2023]. Class unlearning randomly remaps the feature labels of the target class to several classes. Sample unlearning is implemented via backdoor attacks: fixed triggers are injected into the bottom-right pixels of images or appended as specific trigger tokens to the end of text sequences. During the unlearning process, all triggered samples are consistently predicted as class 0, thereby yielding accuracy of class 0 (0A) on the forgotten samples.

As shown in Table 3, our method achieves superior performance across a wide range of datasets, model architectures, and unlearning scenarios. Among them, CIFAR-100 is the most challenging benchmark, while MNIST is the least. Benefiting from the pluggable filters, FedUP reduces the accuracy on the unlearning knowledge to a minimal level (F-A) while

Dataset \ Scenarios		Client Unlearning						Class Unlearning				Sample Unlearning		
		E-C	Federaser	E-F	FUSED	Retrain	FedUP	FUSED	Retrain	FedUP		FUSED	Retrain	FedUP
MNIST-LeNet5	*R-A*	0.99	0.97	0.97	0.99	0.99	**0.99**	0.97	0.99	**1.00**	*0A*	0.95	**1.00**	0.95
	F-A	0.00	0.00	0.00	0.00	0.00	**0.00**	0.00	0.00	**0.00**	*PS*	1.00	1.00	**1.00**
	MIA	0.80	**0.60**	0.76	0.69	0.67	0.70	0.99	**0.97**	0.99	*MIA*	0.96	**0.68**	0.97
CIFAR10-ResNet18	*R-A*	0.57	0.46	0.56	0.56	**0.58**	0.57	0.71	**0.72**	0.71	*0A*	0.60	**0.73**	0.70
	F-A	0.04	0.08	0.04	0.05	0.03	**0.03**	0.00	0.00	**0.00**	*PS*	0.52	**0.60**	0.53
	MIA	0.67	0.70	0.62	0.68	**0.57**	0.63	0.81	**0.63**	0.78	*MIA*	**0.95**	0.98	0.96
CIFAR10-Transformer	*R-A*	0.48	0.51	0.50	0.49	0.52	**0.52**	0.59	**0.60**	0.59	*0A*	0.35	**0.64**	0.57
	F-A	0.05	0.05	0.05	0.05	0.04	**0.04**	0.00	0.00	**0.00**	*PS*	0.50	**0.63**	0.52
	MIA	0.54	**0.35**	0.59	0.56	0.64	0.52	0.81	**0.78**	0.90	*MIA*	0.96	0.91	**0.76**
AG News-TinyBert	*R-A*	0.89	0.88	0.88	0.89	0.89	**0.89**	0.89	0.93	**0.93**	*0A*	0.92	0.93	**0.93**
	F-A	**0.03**	0.05	0.04	0.04	0.03	0.05	0.00	0.00	**0.00**	*PS*	0.87	**0.89**	0.88
	MIA	**0.63**	0.63	0.79	0.68	0.68	0.77	**0.44**	0.63	0.50	*MIA*	0.77	**0.54**	0.70
CIFAR100-ResNet34	*R-A*	0.13	0.15	0.27	0.26	**0.27**	0.26	0.36	0.37	**0.37**	*0A*	0.50	**0.57**	0.55
	F-A	0.01	0.01	0.01	0.01	**0.01**	0.02	0.00	0.00	**0.00**	*PS*	0.55	**0.56**	0.54
	MIA	0.32	0.49	0.43	0.40	**0.26**	0.69	0.91	**0.38**	0.67	*MIA*	0.98	**0.98**	0.99

Table 3: Main results. Our method achieves the best or near-best R-A in most settings while maintaining low F-A, and it performs particularly well in the class unlearning scenario, indicating that it maximizes model utility while ensuring effective unlearning.

maintaining high accuracy on the retained data (R-A). Owing to its centroid-based unlearning mechanism, FedUP exhibits particularly strong performance in class unlearning scenarios. Overall, the proposed method supports single-round communication and reversible unlearning, achieving the desirable triad of high retention accuracy, low forgetting accuracy, and privacy guarantees across all evaluated datasets.

Analysis of Non-target Knowledge Loss. To validate the effectiveness of our method in mitigating non-target knowledge loss, we compare the accuracy on retained knowledge before and after executing the unlearning operation. As shown in Figure 4, FedUP achieves an effect comparable to the Retrain method, characterized by minimal non-target knowledge loss and stable model performance before and after the unlearning operation.

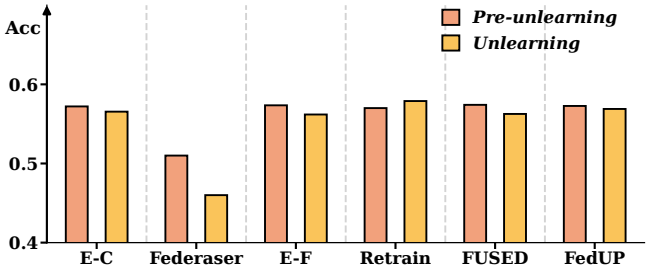


Figure 4: Non-target knowledge loss of methods.

Unlearning Response Time. As shown in Table 4, when achieving the desired forgetting effect, our method requires only 6.33 seconds for image unlearning and 6.45 seconds for text unlearning, which is less than 10% of the latency of the second-best method and less than 1% of that of the Retrain approach. It can promptly respond to diverse unlearning requests across different modalities while maintaining consistently low latency, effectively minimizing processing delays.

Communication Cost. Communication cost is defined as the wireless resources consumed for transmitting model parameters or gradients between local clients and the central server. As shown in Table 4, although the centroid uploading phase introduces additional communication overhead, FedUP requires only a single communication round, meaning its communication cost does not accumulate with rounds. In contrast, model transmission methods gradually converge as rounds increase, causing the communication overhead to escalate to the order of 10^3. Specifically, the lightweight filter incurs an overhead of merely 11.50MB for image unlearning, approximately one-third of that of the second-best method and 6.73MB for text unlearning, which is second only to the FUSED method.

Dataset \ Methods	CIFAR10		AG News	
	Time	Comm	Time	Comm
E-C	1627.76	1110.98	567.56	268.64
Federaser	1586.59	2820.18	634.67	738.76
E-F	898.70	1452.82	629.83	369.38
FUSED	151.21	31.36	88.31	**0.55**
Retrain	935.54	1538.25	859.28	537.28
FedUP	**6.33**	**11.50**	**6.45**	6.73

Table 4: Comparing time and communication costs.

4.3 Analysis of Hyper-parameters

Bottleneck Dimension. In this section, we analyse the dimension of the filter. The pluggable filter adopts a straightforward encoder-decoder architecture. Its input dimension is configured to match the output dimension of the feature extractor. Our empirical investigation focuses on determining the optimal bottleneck dimension within this encoder-decoder structure. As shown in Table 5, the remember accuracy consistently reaches its optimum across both image and text datasets when the bottleneck dimension is set to 32.

Sensitivity Analysis of the Loss Function. The filter is

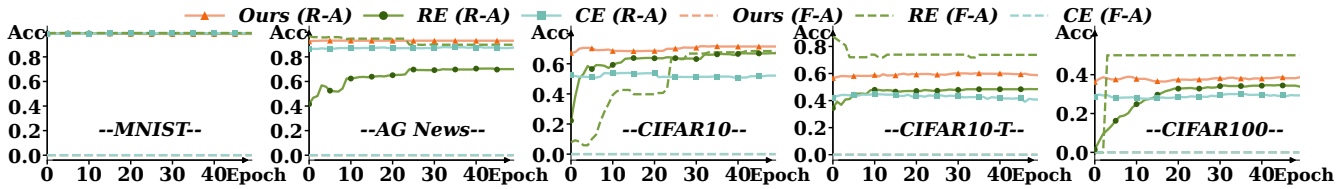


Figure 5: Ablation of loss functions.

Dimension / Dataset	4		8		16		32		64	
	R-A	M	R-A	M	R-A	M	R-A	M	R-A	M
MNIST	0.98	16	0.99	32	0.99	64	**0.99**	128	0.99	256
CIFAR10	0.55	16	0.58	32	0.58	64	**0.60**	128	0.59	256
CIFAR10-T	0.43	16	0.47	32	0.46	64	**0.51**	128	0.46	256
AG News	0.89	4	0.90	8	0.90	16	**0.91**	32	0.90	64
CIFAR100	0.06	16	0.18	32	0.24	64	**0.25**	128	0.23	256

Table 5: Exploration of filter. "M" stands for memory (KB).

trained with joint reconstruction and cross-entropy losses. We performed a sensitivity analysis on α of $\mathcal{L}_{\text{total}}$ to determine its optimal value. As shown in Figure 5, using either loss alone hinders convergence and degrades the accuracy of retained knowledge, whereas their combination maximizes the reduction of non-target knowledge loss. This is further illustrated in Table 6, which shows that increasing the weight of the cross-entropy loss lowers the recognition accuracy on retained data, while assigning a higher weight to the reconstruction loss deteriorates the unlearning effectiveness.

α-CE / metric	0.10		0.30		0.50		0.70		0.90	
	R-A	F-A	R-A	F-A	R-A	F-A	R-A	F-A	R-A	F-A
MNIST	0.99	0.67	**1.00**	0.00	0.99	0.00	0.99	0.00	0.99	0.00
CIFAR10	0.70	0.00	0.68	0.00	**0.71**	0.00	0.68	0.00	0.64	0.00
CIFAR10-T	0.51	0.22	0.56	0.00	**0.59**	0.00	0.51	0.00	0.45	0.00
AG News	0.92	0.00	0.93	0.00	**0.93**	0.00	0.92	0.00	0.87	0.00
CIFAR100	0.35	0.11	0.37	0.00	**0.37**	0.00	0.31	0.00	0.30	0.00

Table 6: Impact of α-CE on R-A and F-A. The filter shows optimal and well-balenced performance at $\alpha = 0.5$.

4.4 Ablation Study

Analysis of Differential Privacy Effects. To ensure privacy preservation, Gaussian noise is injected into the generated class centroids. We conducted ablation studies to evaluate its impact, as shown in Table 7. The results demonstrate that an appropriately calibrated noise scale does not significantly impede the model's performance on non-target knowledge. Instead, it enhances model robustness, achieving a synergistic improvement in both privacy protection and model utility.

Effectiveness of Class Centroid Samples. Class centroids are obtained by clustering original features and privately aggregated on the server side. To validate the effectiveness of class centroid samples, we additionally conduct experiments in which the filter is trained using only the original features. As illustrated in Table 8, the model trained on class centroid samples achieves comparable performance on the accuracy of retained and unlearning knowledge to that trained on the original features, demonstrating that class centroid samples are as effective as the original features.

Dataset / Metrics	CIFAR10		AG News	
	DP	w/o DP	DP	w/o DP
R-A	0.5814	**0.5833**	**0.9132**	0.9114
F-A	**0.0438**	0.0460	0.0490	**0.0303**
MIA	**0.5239**	0.6373	**0.5654**	0.6196

Table 7: Differential privacy effects.

Acc / Dataset	R-A		F-A	
	Feature	Centroid	Feature	Centroid
MNIST	0.9900	**0.9921**	**0.0001**	0.0005
CIFAR10	0.6011	**0.6020**	0.0498	**0.0331**
CIFAR10-T	0.5264	**0.5671**	0.0632	**0.0490**
AG News	**0.9153**	0.9118	**0.0331**	0.0482
CIFAR100	**0.2537**	0.2508	**0.0076**	0.0172

Table 8: Ablation of class centroid samples.

5 Conclusion and Discussion

Conclusion. In the field of FU, server-side methods face non-target knowledge loss, whereas client-side methods incur high request latency. Both are limited by the irreversibility of the unlearning operation. To address these issues, this paper proposes a one-shot federated unlearning framework. By employing differentially private class centroid samples on the server, our approach achieves approximate unlearning that surpasses exact unlearning in effect, reducing non-target knowledge loss and high resource overhead. Through fine-tuning a lightweight plug-in filter in a single round, the desired unlearning effect is achieved, significantly reducing the latency of unlearning responses. Removing the filter allows the model to revert to its pre-unlearning state, thus realizing the reversibility of unlearning. Extensive experiments across diverse datasets, scenarios, and models demonstrate that FedUP demonstrates excellent performance.

Discussion. While FedUP advances unlearning efficiency, reversibility, and responsiveness, two fundamental limitations still persist: Class centroid fidelity depends on federated feature extraction. Low-quality global models propagate bias into aggregated class centroids. It is expected that these limitations can be effectively addressed by utilizing more powerful pre-trained backbones for local feature extraction to ensure reliable class centroids.

Contribution Statement
Feihong Nan and Zhengyi Zhong contributed equally.

References

[Bourtoule *et al.*, 2021] Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE symposium on security and privacy (SP)*, pages 141–159. IEEE, 2021.

[Cao *et al.*, 2018] Yinzhi Cao, Alexander Fangxiao Yu, Andrew Aday, Eric Stahl, Jon Merwine, and Junfeng Yang. Efficient repair of polluted machine learning systems via causal unlearning. In *ASIACCS '18: Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, 2018.

[Chen *et al.*, 2022] Min Chen, Zhikun Zhang, Tianhao Wang, Michael Backes, Mathias Humbert, and Yang Zhang. Graph unlearning. In *Proceedings of the 2022 ACM SIGSAC conference on computer and communications security*, pages 499–513, 2022.

[de la Torre, 2018] Lydia de la Torre. A guide to the california consumer privacy act of 2018. *SSRN Electronic Journal*, Dec 2018.

[Deng *et al.*, 2024] Zhipeng Deng, Luyang Luo, and Hao Chen. Enable the right to be forgotten with federated client unlearning in medical imaging. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 240–250. Springer, 2024.

[Fu *et al.*, 2025a] Lele Fu, Sheng Huang, Yanyi Lai, Chuanfu Zhang, Hong-Ning Dai, Zibin Zheng, and Chuan Chen. Federated domain-independent prototype learning with alignments of representation and parameter spaces for feature shift. *IEEE Transactions on Mobile Computing*, 2025.

[Fu *et al.*, 2025b] Lele Fu, Sheng Huang, Yuecheng Li, Chuan Chen, Chuanfu Zhang, and Zibin Zheng. Learn the global prompt in the low-rank tensor space for heterogeneous federated learning. *Neural Networks*, 187:107319, 2025.

[Golatkar *et al.*, 2020] Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9304–9312, 2020.

[Graves *et al.*, 2021] Laura Graves, Vineel Nagisetty, and Vijay Ganesh. Amnesiac machine learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11516–11524, 2021.

[Guo *et al.*, 2019] Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens Van Der Maaten. Certified data removal from machine learning models. *arXiv preprint arXiv:1911.03030*, 2019.

[Gupta *et al.*, 2021] Varun Gupta, Christopher Jung, Seth Neel, Aaron Roth, Saeed Sharifi-Malvajerdi, and Chris Waites. Adaptive machine unlearning. *Advances in Neural Information Processing Systems*, 34:16319–16330, 2021.

[Halimi *et al.*, 2022] Anisa Halimi, Swanand Kadhe, Ambrish Rawat, and Nathalie Baracaldo. Federated unlearning: How to efficiently erase a client in fl? Jul 2022.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[Huynh *et al.*, 2025] Thanh Trung Huynh, Trong Bang Nguyen, Thanh Toan Nguyen, Phi Le Nguyen, Hongzhi Yin, Quoc Viet Hung Nguyen, and Thanh Tam Nguyen. Certified unlearning for federated recommendation. *ACM Transactions on Information Systems*, 43(2):1–29, 2025.

[Jiang *et al.*, 2026] Wenzheng Jiang, Ke Liang, Wenke Huang, Xiongtao Zhang, Zhenxing Xu, Guancheng Wan, Cheston Tan, Flint Xiaofeng Fan, and Ji Wang. Unveiling and mitigating untargeted poisoning attacks on federated knowledge graph embedding. In *Proceedings of the ACM Web Conference 2026*, pages 2569–2580, 2026.

[Jiao *et al.*, 2020] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling bert for natural language understanding. In *Findings of the association for computational linguistics: EMNLP 2020*, pages 4163–4174, 2020.

[Krizhevsky *et al.*, 2009] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[Kuo *et al.*, 2025] Kevin Kuo, Amrith Setlur, Kartik Srinivas, Aditi Raghunathan, and Virginia Smith. Exact unlearning of finetuning data via model merging at scale. *arXiv preprint arXiv:2504.04626*, 2025.

[Kurmanji *et al.*, 2023] Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. Towards unbounded machine unlearning. *Advances in neural information processing systems*, 36:1957–1987, 2023.

[LeCun *et al.*, 2002] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 2002.

[Li *et al.*, 2022] Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. Federated learning on non-iid data silos: An experimental study. In *2022 IEEE 38th international conference on data engineering (ICDE)*, pages 965–978. IEEE, 2022.

[Li *et al.*, 2025] Na Li, Chunyi Zhou, Yansong Gao, Hui Chen, Zhi Zhang, Boyu Kuang, and Anmin Fu. Machine unlearning: Taxonomy, metrics, applications, challenges, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*, 2025.

[Liu *et al.*, 2021] Gaoyang Liu, Xiaoqiang Ma, Yang Yang, Chen Wang, and Jiangchuan Liu. Federaser: Enabling efficient client-level data removal from federated learning models. In *2021 IEEE/ACM 29th International Symposium on Quality of Service (IWQOS)*, pages 1–10. IEEE, 2021.

[Liu *et al.*, 2022] Yi Liu, Lei Xu, Xingliang Yuan, Cong Wang, and Bo Li. The right to be forgotten in federated learning: An efficient realization with rapid retraining. In *IEEE INFOCOM 2022-IEEE conference on computer communications*, pages 1749–1758. IEEE, 2022.

[Liu *et al.*, 2024] Ziyao Liu, Yu Jiang, Jiyuan Shen, Minyi Peng, Kwok-Yan Lam, Xingliang Yuan, and Xiaoning Liu. A survey on federated unlearning: Challenges, methods, and future directions. *ACM Computing Surveys*, 57(1):1–38, 2024.

[Ma *et al.*, 2022] Zhuo Ma, Yang Liu, Ximeng Liu, Jian Liu, Jianfeng Ma, and Kui Ren. Learn to forget: Machine unlearning via neuron masking. *IEEE Transactions on Dependable and Secure Computing*, 20(4):3194–3207, 2022.

[McMahan *et al.*, 2016] H.Brendan McMahan, EiderB Moore, Daniel Ramage, Seth Hampson, and BlaiseAgüeray Arcas. Communication-efficient learning of deep networks from decentralized data. *arXiv: Learning,arXiv: Learning*, Feb 2016.

[Mora *et al.*, 2024] Alessio Mora, Luca Dominici, and Paolo Bellavista. Fedunran: On-device federated unlearning via random labels. In *2024 IEEE International Conference on Big Data (BigData)*, pages 7955–7960. IEEE, 2024.

[Nguyen *et al.*, 2025] 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. *ACM Transactions on Intelligent Systems and Technology*, 16(5):1–46, 2025.

[Pan *et al.*, 2025] Zibin Pan, Zhichao Wang, Chi Li, Kaiyan Zheng, Boqi Wang, Xiaoying Tang, and Junhua Zhao. Federated unlearning with gradient descent and conflict mitigation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 19804–19812, 2025.

[Qi *et al.*, 2023] Zhuang Qi, Lei Meng, Zitan Chen, Han Hu, Hui Lin, and Xiangxu Meng. Cross-silo prototypical calibration for federated learning with non-iid data. In *Proceedings of the 31st ACM international conference on multimedia*, pages 3099–3107, 2023.

[Qi *et al.*, 2024] Zhuang Qi, Lei Meng, Weihao He, Ruohan Zhang, Yu Wang, Xin Qi, and Xiangxu Meng. Cross-training with multi-view knowledge fusion for heterogenous federated learning. *arXiv e-prints*, pages arXiv–2405, 2024.

[Truong *et al.*, 2021] Nguyen Truong, Kai Sun, Siyao Wang, Florian Guitton, and YiKe Guo. Privacy preservation in federated learning: An insightful survey from the gdpr perspective. *Computers & Security*, 110:102402, 2021.

[Vaswani *et al.*, 2017] 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, 2017.

[Wang *et al.*, 2022] Junxiao Wang, Song Guo, Xin Xie, and Heng Qi. Federated unlearning via class-discriminative pruning. In *Proceedings of the ACM web conference 2022*, pages 622–632, 2022.

[Wang *et al.*, 2023a] Cheng-Long Wang, Mengdi Huai, and Di Wang. Inductive graph unlearning. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 3205–3222, 2023.

[Wang *et al.*, 2023b] Weiqi Wang, Zhiyi Tian, Chenhan Zhang, An Liu, and Shui Yu. Bfu: Bayesian federated unlearning with parameter self-sharing. In *Proceedings of the 2023 ACM Asia Conference on Computer and Communications Security*, pages 567–578, 2023.

[Wei *et al.*, 2020] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H Yang, Farhad Farokhi, Shi Jin, Tony QS Quek, and H Vincent Poor. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE transactions on information forensics and security*, 15:3454–3469, 2020.

[Wu *et al.*, 2022] Chen Wu, Sencun Zhu, and Prasenjit Mitra. Federated unlearning with knowledge distillation. *arXiv preprint arXiv:2201.09441*, 2022.

[Xiong *et al.*, 2023] Zuobin Xiong, Wei Li, Yingshu Li, and Zhipeng Cai. Exact-fun: an exact and efficient federated unlearning approach. In *2023 IEEE International Conference on Data Mining (ICDM)*, pages 1439–1444. IEEE, 2023.

[Yang *et al.*, 2025] Zhe-Rui Yang, Jindong Han, Chang-Dong Wang, and Hao Liu. Erase then rectify: A training-free parameter editing approach for cost-effective graph unlearning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 13044–13051, 2025.

[Zhang *et al.*, 2015] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

[Zhong *et al.*, 2022] Zhengyi Zhong, Weidong Bao, Ji Wang, Xiaomin Zhu, and Xiongtao Zhang. Flee: A hierarchical federated learning framework for distributed deep neural network over cloud, edge, and end device. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 13(5):1–24, 2022.

[Zhong *et al.*, 2025a] Zhengyi Zhong, Weidong Bao, Ji Wang, Jianguo Chen, Lingjuan Lyu, and Wei Yang Bryan Lim. Sacfl: Self-adaptive federated continual learning for resource-constrained end devices. *IEEE Transactions on Neural Networks and Learning Systems*, 2025.

[Zhong *et al.*, 2025b] Zhengyi Zhong, Weidong Bao, Ji Wang, Shuai Zhang, Jingxuan Zhou, Lingjuan Lyu, and Wei Yang Bryan Lim. Unlearning through knowledge overwriting: Reversible federated unlearning via selective sparse adapter. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 30661–30670, 2025.

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