Unlearning during Learning: An Efficient Federated Machine Unlearning Method

Hanlin Gu[1] , **Gongxi Zhu**[2,3] , **Jie Zhang**[4] , **Xinyuan Zhao**[3] ,
Yuxing Han[3] , **Lixin Fan**[1] and **Qiang Yang**[1]
[1]AI Lab, Webank ,
[2] University of Electronic Science Technology of China ,
[3] Shenzhen International Graduate School, Tsinghua University ,
[4] Nanyang Technological University
allengu@webank.com, gx.zhu@foxmail.com, jie_zhang@ntu.edu.sg, yuxinghan@sz.tsinghua.edu.cn

Abstract

In recent years, Federated Learning (FL) has garnered significant attention as a distributed machine learning paradigm. To facilitate the implementation of the "right to be forgotten," the concept of federated machine unlearning (FMU) has also emerged. However, current FMU approaches often involve additional time-consuming steps and may not offer comprehensive unlearning capabilities, which renders them less practical in real FL scenarios. In this paper, we introduce FedAU, an innovative and efficient FMU framework aimed at overcoming these limitations. Specifically, FedAU incorporates a lightweight auxiliary unlearning module into the learning process and employs a straightforward linear operation to facilitate unlearning. This approach eliminates the requirement for extra time-consuming steps, rendering it well-suited for FL. Furthermore, FedAU exhibits remarkable versatility. It not only enables multiple clients to carry out unlearning tasks concurrently but also supports unlearning at various levels of granularity, including individual data samples, specific classes, and even at the client level. We conducted extensive experiments on MNIST, CIFAR10, and CIFAR100 datasets to evaluate the performance of FedAU. The results demonstrate that FedAU effectively achieves the desired unlearning effect while maintaining model accuracy.

1 Introduction

Federated learning (FL) [Konečnỳ *et al.*, 2015; McMahan *et al.*, 2017; Yang *et al.*, 2019] is a promising distributed machine learning paradigm that provides privacy-preserving learning solutions. One essential requirement of FL is the participants' "right to be forgotten", which has been stated explicitly in the European Union General Data Protection Regulation (GDPR)[1] and the California Consumer Privacy Act (CCPA) [Harding *et al.*, 2019]. Federated Machine Unlearning (FMU) is proposed to give clients the right to remove the influence of a certain subset of their data from a trained federated learning (FL) model, while maintaining the accuracy of the FL model on remaining data [Che *et al.*, 2023].

Three representative existing FMU approaches have been proposed. Firstly, One prevalent approach involves the retraining or fine-tuning of the model from scratch using the *remaining data*[Liu *et al.*, 2022a; Liu *et al.*, 2021; Su and Li, 2023; Zhang *et al.*, 2023]. Secondly, another line of research explores the utilization of Gradient Ascent on the *unlearning data* to effectively diminish its impact [Wu *et al.*, 2022; Graves *et al.*, 2021]. Thirdly, [Wang *et al.*, 2022] explored the application of model pruning techniques. Specifically, they selectively removed certain neurons from the model architecture that exhibit a high correlation with the unlearning data. In practice, there are two important ingredients required for FMU [Zhang *et al.*, 2023; Liu *et al.*, 2023]:

- **Reduced Unlearning Time**: FL systems require FMU methods that minimize the time required for unlearning operations. This is crucial because normal clients participating in FL cannot afford to wait for the unlearning client to complete the unlearning process. Even for methods like gradient ascent [Wu *et al.*, 2022; Graves *et al.*, 2021] and pruning [Wang *et al.*, 2022], there is still a need for a certain amount of time to implement the unlearning operation.
- **Broad Unlearning Capability**: An effective FMU method should have the capability to accommodate unlearning requests from multiple clients in FL. This includes the ability to unlearn specific samples, classes, or clients as requested by different clients participating in the FL process.

However, existing methods do not consider these two important requirements simultaneously. In order to satisfy these two requirements, we propose an efficient Federated Machine Unlearning (FMU) framework called FedAU. FedAU incorporates an auxiliary unlearning module during the training that facilitates the unlearning process. Our framework offers three key advantages: Firstly, FedAU utilizes a simple linear operation to achieve unlearning, which avoids consuming the waiting time for other normal clients during the federated learning process (see Sect. 3.2). Secondly, FedAU is a general unlearning framework that allows multiple clients to implement unlearning. It supports unlearning at the sample,

[1]https://gdpr-info.eu/art-17-gdpr/

class, and client levels, providing flexibility in managing privacy concerns (see Sect. 3.3). Thirdly, the proposed FedAU demonstrates strong performance in terms of unlearning effectiveness and model accuracy. This is supported by both theoretical analysis and experimental evaluations (see Sect. 3.4 and Sect. 4).

Contribution. The main contributions are summarized as follows:

- We point out that existing methods for federated machine unlearning (FMU) is not feasible in practice, in terms of unlearning time and unlearning capability.
- In this paper, we propose FedAU, a streamlined FMU framework, that incorporates a lightweight auxiliary unlearning module into the learning process and adopts a linear operation to achieve unlearning.
- Extensive experiments and theoretical analysis demonstrated that FedAU is highly effective in enabling unlearning across various scenarios.

2 Relate Work

2.1 Federated Learning

Federated learning [Konečnỳ *et al.*, 2015; McMahan *et al.*, 2017; Cheng *et al.*, 2020] aims to build a machine learning model based on datasets that are distributed across multiple devices without sharing private data with the server and other devices. The cornerstone of federated learning algorithms, FedAvg, proposed by [McMahan *et al.*, 2017], involves local clients training models on their data and sending model updates to a central server. The server then averages these updates to improve a global model. Afterward, researchers have proposed various optimization techniques [Deng *et al.*, 2020; Sun *et al.*, 2022] to enhance FedAvg.

Given the privacy-centric nature of federated learning, a plethora of research focuses on enhancing data privacy. Techniques such as differential privacy [Dwork, 2006] and secure multi-party computation [Goldreich, 1998] are often integrated into federated learning algorithms to protect client data. Nevertheless, recent research has highlighted vulnerabilities in federated learning to privacy breaches, notably through model inversion attacks [Nasr *et al.*, 2019] and membership inference attacks [He *et al.*, 2019]. In this paper, we focus on leveraging unlearning techniques to mitigate these privacy risks in federated learning scenarios. Notably, we utilize FedAvg as the default federated learning algorithm.

2.2 Machine Unlearning

Machine unlearning [Bourtoule *et al.*, 2021; Mercuri *et al.*, 2022] involves removing the influence of specific training data from a machine learning model, often for privacy, fairness, or data quality reasons. It is a response to challenges like the "right to be forgotten" in the context of data privacy regulations. A pivotal advancement in machine unlearning is the development of a unified PAC-Bayesian framework [Jose and Simeone, 2021], which recasts variational unlearning and forgetting Lagrangian as information risk minimization problems. Another significant development is the introduction of cryptographic frameworks for verifiable machine unlearning [Eisenhofer *et al.*, 2022], which allows users to verify the removal of their data.

The application of machine unlearning in federated learning environments presents unique challenges and opportunities [Liu *et al.*, 2022b]. Traditional methods, such as retraining models on remaining data [Liu *et al.*, 2022a; Su and Li, 2023] or directly modifying the original model [Liu *et al.*, 2021; Halimi *et al.*, 2022; Wu *et al.*, 2022], are often too time-intensive to be viable in the dynamic setting of federated learning. Furthermore, the federated learning paradigm involves multiple clients, each potentially requesting data unlearning. This scenario adds complexity, as current methodologies rarely address the efficient unlearning of data from numerous clients simultaneously. Additionally, while some methods utilize noise addition for efficiency [Sekhari *et al.*, 2021; Zhang *et al.*, 2023], this approach can compromise the performance of models trained on the remaining data, leading to a trade-off between efficiency and model accuracy. In this paper, our goal is to develop a streamlined approach to federated machine unlearning, adaptable across a range of applications.

3 The Proposed Method

We introduce the FMU setting in Sect. 3.1, followed by elaboration on the proposed FMU framework, FedAU, in Sect. 3.2. Then we show the generality of FedAU in Sect. 3.3: 1) it can be applied in unlearning sample, class, and client; 2) it allows multiple clients to request to unlearn. Finally, we provide a theoretical analysis of the influence of FedAU on model accuracy and the unlearning effect. in Sec. 3.4.

3.1 Setting

Consider a Horizontal Federated Learning setting consisting of K clients who collaboratively train a FL model $\omega = (E, W)$ (Feature extractor E and Classifier W) to optimize the following objective:

$$\min_{\omega} \sum_{k=1}^{K} \sum_{i=1}^{n_k} \frac{\ell(F_{E,W}(x_{k,i}), y_{k,i})}{n_1 + \cdots + n_K}, \tag{1}$$

where ℓ is the loss, e.g., the cross-entropy loss and $\mathcal{D}_k = \{(x_{k,i}, y_{k,i})\}_{i=1}^{n_k}$ is the dataset with size n_k owned by client k. Client k_0 requests to withdraw their consent for the utilization of its data, resulting in the need for the server to remove the influence of the data contributed by clients k_0 from the trained model. Moreover, if multiple clients request to unlearn simultaneously, we define the set of these multiple clients to be $\mathcal{C}$.

We note three distinct cases within the Federated Model Unlearning (FMU) framework

1. *Unlearning samples*: In this case, the goal is to eliminate the knowledge acquired from a subset of client data, thereby excluding it from the global model.
2. *Unlearning class*: This case involves excluding a specific class from the model's generalization boundary, effectively removing it from the model's predictions.
3. *Unlearning client*: Here, the objective is to completely erase the data of a particular client, denoted as $\mathcal{D}_{k_0}^{u} =$

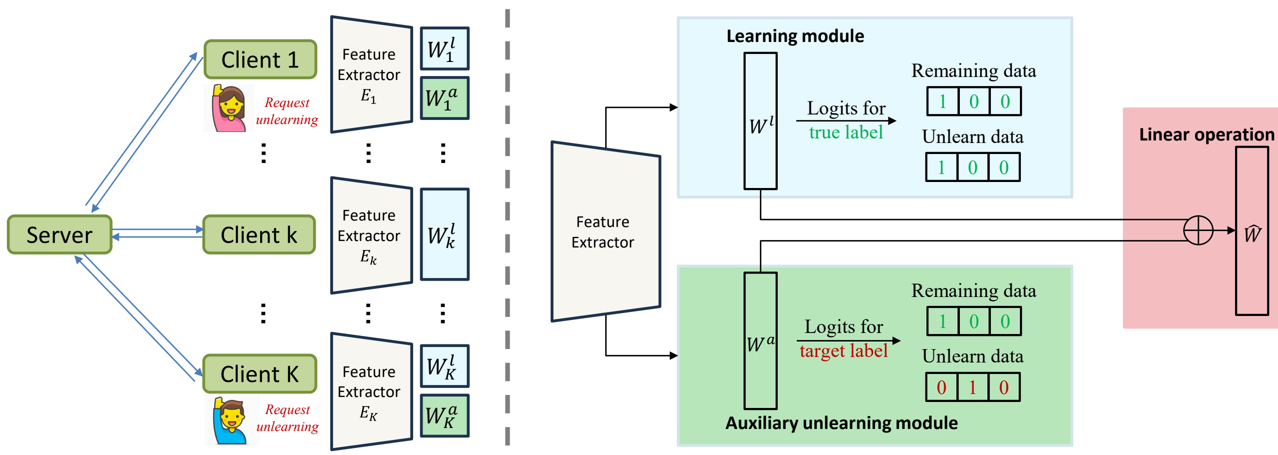


Figure 1: Left: the scenario of federated machine unlearning; Right: the overview of the proposed FedAU consisted of three modules/operations (blue: learning module, green: auxiliary unlearning module and red: linear operation).

D_{k_0}, ensuring that the model is no longer influenced by any data from that client. The detailed experiment to analyze the unlearning effect for the the number of unlearning samples $|\mathcal{D}^u_{k_0}|$ is illustrated in Appendix C.

3.2 FedAU

As shown in the right of Figure 1, our main approach involves the incorporation of an auxiliary unlearning module W^a during the training process of the dataset. When a client requests to unlearn specific information, a straightforward linear operation, such as a weighted average, can be taken between the learning module W^l and the auxiliary unlearning module W^a to produce the final unlearning model $\hat{W}$. More details are provided below.

Learning Module

Consider the task of supervised classification using Deep Neural Networks (DNNs). Let $\mathcal{Y} = \{1, ..., C\}$ denote the label space, where C represents the total number of classes. The learning module aims to optimize the following objective:

$$E, W^l = \underset{E,W}{\operatorname{argmin}} \sum_{k=1}^{K} \sum_{(x_{k,i}, y_{k,i}) \in \mathcal{D}_k} \frac{\ell(F_{E,W}(x_{k,i}), y_{k,i})}{n_1 + \cdots + n_K}. \tag{2}$$

Here, ℓ represents the loss function, such as the cross-entropy loss, and $\mathcal{D}_k = \{(x_{k,i}, y_{k,i})\}_{i=1}^{n_k}$ represents the dataset of client k with a size of n_k.

Auxiliary Unlearning Module

We design an auxiliary unlearning module W^a that is learned by clients $\mathcal{C}$ who request to unlearn their data. The goal of the auxiliary unlearning module is to learn a special model $W^a_{k_0}$ for the designed data D'_{k_0} of client k_0 as:

$$W^a_{k_0} = \underset{W}{\operatorname{argmin}} \sum_{(x_{k_0,i}, y_{k_0,i}) \in \mathcal{D}'_{k_0}} \frac{\ell(F_{E,W}(x_{k_0,i}), y_{k_0,i})}{|\mathcal{D}'_{k_0}|}. \tag{3}$$

Then we can implement the simple linear operation between W^l and $W^a_{k_0}$ in the following section to obtain the unlearning model $\hat{W}$, which can remove the influence of the unlearning data $D^u_{k_0}$. The auxiliary unlearning module has two characteristics: 1) it is trained during the learning process and efficiently converge with the several training epoch when initializing as W^l; 2) multiple unlearning clients $\mathcal{C}$ can train their own auxiliary unlearning privately or collaboratively to deal with different condition (see unlearning sample in Sec. 3.3).

Linear Operation on W^l and W^a

The unlearning model $\hat{W}$ needs to satisfy two requirements for unlearning data D^u and remaining data $\mathcal{D}^r = \mathcal{D} - \mathcal{D}^u$. The ***first requirement*** is that unlearning doesn't influence the model accuracy of the remaining data $\mathcal{D}^r$. Specifically, the logit output of $\hat{W}$ represents the same to the W^l w.r.t the remaining data $\mathcal{D}^r$, i.e.,

$$\underset{i}{\operatorname{argmax}}\, F^i_{E,W^l}(x) = \underset{i}{\operatorname{argmax}}\, F^i_{E,\hat{W}}(x), x \in \mathcal{D}^r, \tag{R1}$$

where $F_{E,W}(x)$ is the logit output with size C by the trained model on the input x and $F^i_{E,W}(x)$ is the i_{th} logit. The ***second requirement*** is that the model after unlearning behaves wrongly the unlearning data $\mathcal{D}^u$ such as [Chen *et al.*, 2023]. Specifically, it requires the logit output of $\hat{W}$ shows the difference to the unlearning data $\mathcal{D}^u$, i.e.,

$$\underset{i}{\operatorname{argmax}}\, F^i_{E,\hat{W}}(x) \neq y, x \in \mathcal{D}^u, \tag{R2}$$

where y is true label of x. In other words, this requirement indicates that the model after unlearning doesn't memorize the unlearning data $\mathcal{D}^u$.

Remark 1. *Some methods [Graves* et al.*, 2021] aims to achieve the requirement* (R2) *by finetuning the trained model with randomly labeled forgetting data, but this will also shift the boundary of the remaining class randomly, leading to the degeneration of utility the on remaining data.*

To make the simple linear operation as described by Eq. (6) and concurrently to satisfy the aforementioned two requirements ((R1) and (R2)), we leverage the linear property of a fully connected layer to achieve this:

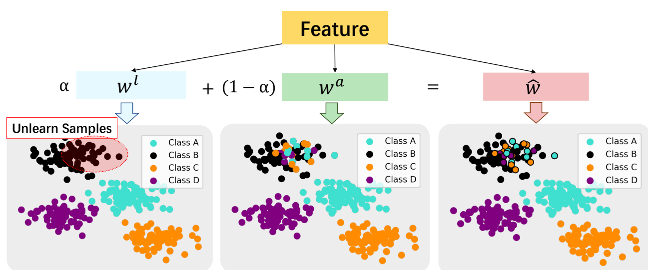
$$\hat{W} = W^l \bigoplus W^a. \tag{6}$$

The following proposition illustrates that the change in logits is proportional to the change in weights if the network is fully connected layer:

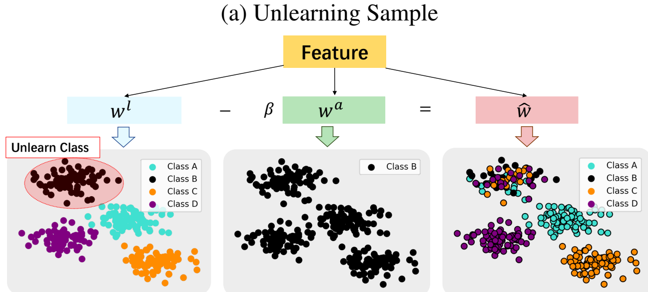
Proposition 1. *Consider two fully connected layers projecting the input $x \in \mathbb{R}^{m_2}$ to the logit $l \in \mathbb{R}^{m_1}$ as: $l_1 = w_1 x + b_1, l_2 = w_2 x + b_2$, then the linear operation of weights w_1, b_1 and w_2, b_2 has the same influence on logits l_1 and l_2.*

Therefore, by utilizing this property, the model change can effectively reflect the change in logits, thereby satisfying the aforementioned requirements. The specific design of this linear operation for unlearning samples and classes are introduced in the following section.

Remark 2. *There is no need to train Auxiliary unlearning module at the beginning of learning. As indicated in Appendix C, the training of the AU only necessitates a few epochs. Consequently, clients can proactively train the AU module several epochs in advance of the unlearning request rather than at the beginning of learning.*



(a) Unlearning Sample



(b) Unlearning Class

Figure 2: Illustration of the proposed FedAU when unlearning sample and class. After the module W^l undergoes linear operation with the auxiliary unlearning module W^a, the unlearned part of the original feature will be classified into other random classes.

3.3 Generality of FedAU

We provide the details of FedAU on how to unlearn the sample and class in this part (see Alg. 1 and Alg. 2).

Unlearning Sample in FL

We firstly consider the scenario that only one client k_0 attempts to unlearn some samples $\mathcal{D}^u_{k_0} = \{(x^u_{k_0,i}, y^u_{k_0,i})\}^m_{i=1}$, the core steps are as followings:

- Client k_0 *designs an the auxiliary dataset* $\mathcal{D}'_{k_0} = \mathcal{D}^{u'}_{k_0} \cup \mathcal{D}^{r'}_{k_0}$. Specifically, $\mathcal{D}^{u'}_{k_0}$ is based on $\mathcal{D}^u_{k_0}$ by modifying the label $y^u_{k,i}$ with $y^{u'}_{k,i} \sim U(1, C)$ and $\mathcal{D}^{r'}_{k_0} = \mathcal{D}^r_{k_0}$, where $U(1, C)$ represents the discrete uniform distribution on value $1, \cdots, y^u_{k_0,i} - 1, y^u_{k_0,i} + 1, \cdots, C$ (see blue line 3-9 of Algo. 1).
- Then client k_0 *learns the auxiliary unlearning module* $W^a_{k_0}$ according to the $\mathcal{D}'_{k_0}$ during the learning stage (see green line 10-15 of Algo. 1).
- Finally, when the unlearning request is proposed, the unlearning model $\hat{W}$ can be obtained as:

$$\hat{W} = \alpha W^l + (1 - \alpha) W^a_{k_0}, \tag{7}$$

  where α is a small positive coefficient (see red line 21 of Algo. 1).

As shown in Fig. 2(a), the the class of remaining data is not influenced by the addition operation since the remaining and auxiliary dataset have the same class. Moreover, the class of unlearning data is mainly influenced by the auxiliary dataset if $(1 - \alpha)$ tends to 1. Therefore, Linear operation of Eq. (7) satisfies requirements (R1) and (R2), which is also illustrated in Theorem 1.

Unlearning Class in FL

Consider the scenario that only one client k_0 attempts to unlearn class c data as $D^u_{k_0} = \{(x^u_{k_0,i}, c)\}^m_{i=1}$, there are the following three steps in FedAU:

- Client k_0 *designs an the auxiliary dataset* $\mathcal{D}'_{k_0} = \mathcal{D}^{u'}_{k_0} \cup \mathcal{D}^{r'}_{k_0}$. Specifically, $\mathcal{D}^{r'}_{k_0}$ is based on $\mathcal{D}^r_{k_0}$ by modifying the label $y^r_{k_0,i}$ with label c and $\mathcal{D}^{u'}_{k_0} = \mathcal{D}^u_{k_0}$ (see blue line 3-9 of Algo. 1);
- Then client k_0 *learns the auxiliary unlearning module* $W^a_{k_0}$ according to the $\mathcal{D}'_{k_0}$ during the learning stage (see green line 10-15 of Algo. 1).
- Finally, when the unlearning request is proposed, the unlearning model $\hat{W}$ can be obtained as:

$$\hat{W} = W^l - \beta W^a_{k_0}, \tag{8}$$

  where β is a large coefficient (see red line 21 of Algo. 1).

In Fig. 2(b), it can be observed that the subtraction operation does not affect the class of the remaining data. This is because the remaining dataset and the auxiliary dataset have different classes, resulting in the class of the subtrahend being preserved. Additionally, the unlearning data and auxiliary data share the same class (represented by the black point), allowing the class to be removed when the subtraction is performed. Therefore, the linear operation defined by Equation (8) satisfies the requirements (R1) and (R2), as also illustrated in Theorem 1.

Remark 3. *We provide the analysis on how the value of α influence the performance of the remaining data and unlearning data in Sect. 4.3.*

Algorithm 1 Unlearning Sample in FL (Learning Module , Auxiliary Unlearning Module and Linear Operation)

Input: Communication rounds T, Client number K, dataset $\mathcal{D}_{k_0}$ (remaining data $\mathcal{D}^r_{k_0}$ and unlearning data $\mathcal{D}^u_{k_0}$) for unlearning client k_0.

Initialize the feature extractor E, unlearning learning module W^l and auxiliary unlearning module $W^a_{k_0}$
for $t = 1, 2, \dots, T$ **do**
  ▷ *Clients perform:*
  for Client k in $\{1, \dots, K\}$ **do**
    Set $E_k = E$, $W^l_k = W^l$;
    Compute the learning loss $\tilde{\ell} = \ell(\mathcal{D}_k; E_k, W^l_k)$;
    $W^l_k \longleftarrow W^l_k - \eta \nabla_{W^l_k} \tilde{\ell}$;
    $E_k \longleftarrow E_k - \eta \nabla_{E_k} \tilde{\ell}$;
  end for
  Let $W^a_{k_0} = W^a$;
  Set $\mathcal{D}^u_{k_0} = (x^u_{k_0,i}, y^{u'}_{k_0,i} \sim U(1, C))$;
  Set $\mathcal{D}^{r'}_{k_0} = \mathcal{D}^r_{k_0}$;
  Set $\mathcal{D}'_{k_0} = \mathcal{D}^{u'}_{k_0} \cup \mathcal{D}^{r'}_{k_0}$;
  Compute the learning loss $\tilde{\ell} = \ell(\mathcal{D}'_{k_0}; E_{k_0}, W^a_{k_0})$;
  $W^a_{k_0} \longleftarrow W^a_{k_0} - \eta \nabla W^a_{k_0} \tilde{\ell}$;
  Upload the W^l_k and E_k to the server;
  ▷ *The server performs:*
  The server aggregates E and W^l as: $W^l = \frac{1}{K}(W^l_1 + \dots + W^l_K)$; $E = \frac{1}{K}(E_1 + \dots + E_K)$;
  The server distributes E and W^l to all clients.
end for
The server implements unlearning process:

$$\hat{W} = \alpha W^l + (1 - \alpha) W^a_{k_0}$$

return $E, \hat{W}$

Unlearning a Client in FL

Unlearning a client represents an extreme case of unlearning samples, allowing us to leverage strategies used for unlearning samples. The key difference is that in the case of unlearning a client k_0, there is no remaining data from that client ($|\mathcal{D}^r_{k_0}| = 0$). As a result, the auxiliary unlearning module $W^a_{k_0}$ cannot learn from the data of other clients. To address this, we propose an improved updating strategy for the auxiliary unlearning module, which involves combining the knowledge learned from $\mathcal{D}^{u'}_{k_0}$ and the original model W^l for each epoch. Further details can be found in the Appendix B.

Unlearning for Multiple Clients

The proposed FedAU can also be applied into satisfying unlearning request for multiple clients without consuming extra time. Specifically, for unlearning class, each client in $\mathcal{C}$ **privately learn** the $W^a_k, k \in \mathcal{C}$ with the goal of optimizing Eq. (3). Then all clients obtain the unlearning model $\hat{W}$ as:

$$\hat{W} = W^l - \sum_{k \in \mathcal{C}} \beta_k W^a_k.$$

Algorithm 2 Unlearning Class in FL (Learning Module , Auxiliary Unlearning Module and Linear Operation)

Input: Communication rounds T, $\mathcal{D}_{k_0}$ including remaining data $\mathcal{D}^r_{k_0}$ and unlearning data $\mathcal{D}^u_{k_0}$ (the label of $\mathcal{D}^u_{k_0}$ is c) for unlearning client k_0.

Initialize the feature extractor E, unlearning learning module W^l and auxiliary unlearning module $W^a_{k_0}$
for $t = 1, 2, \dots, T$ **do**
  ▷ *Clients perform:*
  for Client k in $\{1, \dots, K\}$ **do**
    Set $E_k = E$, $W^l_k = W^l$;
    Compute the learning loss $\tilde{\ell} = \ell(\mathcal{D}_k; E_k, W^l_k)$;
    $W^l_k \longleftarrow W^l_k - \eta \nabla_{W^l_k} \tilde{\ell}$;
    $E_k \longleftarrow E_k - \eta \nabla_{E_k} \tilde{\ell}$;
  end for
  Let $W^a_{k_0} = W^a$;
  Set $\mathcal{D}^{u'}_{k_0} = \mathcal{D}^u_{k_0}$;
  Set $\mathcal{D}^{r'}_{k_0} = (x^r_{k_0,i}, c)$;
  Set $\mathcal{D}'_{k_0} = \mathcal{D}^{u'}_{k_0} \cup \mathcal{D}^{r'}_{k_0}$;
  Compute the learning loss $\tilde{\ell} = \ell(\mathcal{D}'_{k_0}; E_{k_0}, W^a_{k_0},)$;
  $W^a_{k_0} \longleftarrow W^a_{k_0} - \eta \nabla_{W^a_{k_0}} \tilde{\ell}$;
  Upload the W^l_k and E_k to the server;
  ▷ *The server performs:*
  The server aggregates E and W^l as: $W^l = \frac{1}{K}(W^l_1 + \dots + W^l_K)$; $E = \frac{1}{K}(E_1 + \dots + E_K)$;
  The server distributes E and W^l to all clients.
end for
The server implements unlearning process:

$$\hat{W} = W^l - \beta W^a_{k_0}$$

return $E, \hat{W}$

The detailed algorithm and results shown in Appendix B.

For unlearning sample, multiple clients $\mathcal{C}$ **collaboratively learn** the W^a that aiming to optimize:

$$W^a = \operatorname*{argmin}_{W} \sum_{k \in \mathcal{C}} \sum_{(x_{k,i}, y_{k,i}) \in \mathcal{D}'_k} \frac{\ell(F_{E,W}(x_{k,i}), y_{k,i})}{\sum_{k \in \mathcal{C}} n_k} \tag{9}$$

Then all clients obtain the unlearning model $\hat{W}$ as:

$$\hat{W} = \alpha W^l + (1 - \alpha) W^a$$

The detailed algorithm and results shown in Appendix B.

Remark 4. *In multiple client scenarios, the reason for the difference between unlearning samples and unlearning classes lies in the nature of the linear operations involved. When unlearning a class, the linear operation used is subtraction, which allows for the removal of multiple classes by subtracting W^a_k for each client $k \in \mathcal{C}$ individually. On the contrary, when unlearning samples, the operation is addition, where all W^a_k for each client $k \in \mathcal{C}$ are added together.*

This addition operation can potentially affect the unlearning effect because it is uncertain whether $W^a_{k_1}$ of client k_1 can effectively unlearn the unlearning samples $\mathcal{D}^u_{k_2}$ of client k_2.

3.4 Theoretical Analysis

The following theorem demonstrates the proposed Algorithm 1 and 2 satisfy Requirement R1 and R2 (see proof in Appendix D).

Theorem 1. *For client k_0 aims to remove $\mathcal{D}^u_{k_0}$ from the $\mathcal{D}_{k_0}$, and let $\mathcal{D}^r_{k_0} = \mathcal{D}_{k_0} - \mathcal{D}^u_{k_0}$. There exist α and β such that both unlearning Algorithm 1 and 2 satisfy the requirement* (R1) *and* (R2)*, i.e.,*

$$\begin{cases} \underset{i}{\operatorname{argmax}}\, F^i_{W^l}(x) = \underset{i}{\operatorname{argmax}}\, F^i_{\hat{W}}(x), & x \in \mathcal{D}^r_{k_0}, \\ \underset{i}{\operatorname{argmax}}\, F^i_{\hat{W}}(x) \neq y, & (x, y) \in \mathcal{D}^u_{k_0}, \end{cases} \tag{10}$$

Theorem 1 establishes the effectiveness of FedAU when a single client requests unlearning. Furthermore, we present a comprehensive theoretical analysis of the effectiveness of FedAU in scenarios where multiple clients request unlearning.

4 Experiment

4.1 Experimental Setting

Models & Datasets & Setting. We conduct experiments on three datasets: *MNIST* [LeCun *et al.*, 2010], *CIFAR10* and CIAFR100 [Krizhevsky *et al.*, 2014]. We adopt LeNet [LeCun *et al.*, 1998] for conducting experiments on MNIST and adopt *AlexNet* [Krizhevsky *et al.*, 2012] on CIFAR10 and *ResNet18* [He *et al.*, 2016] on CIFAR100.

We simulate a HFL scenario consisting 10 clients under IID and Non-IID setting [Li *et al.*, 2022] (following the Dirichlet distribution, $dir(\gamma)$). For unlearning samples, we employed the backdoor technique to generate the unlearning samples [Gao *et al.*, 2022]. The proportion of unlearning samples was set to 5%, 10%, and 20% of the dataset. For unlearning a client, we considered scenarios where the data from the unlearning client accounted for 20%, 50%, and 100% of the data from the other clients. In addition, we conducted experiments involving unlearning for multiple clients. We varied the number of unlearning clients, exploring scenarios with 3, 5, 8, and 10 unlearning clients. Furthermore, we treated the last layer of the model as the auxiliary unlearning module. An ablation study on the position of the auxiliary unlearning module is provided in the Appendix B.

The Baseline FMU Methods. We compare six FMU methods, including Retraining/finetuning-based: Retraining, FedEraser [Liu *et al.*, 2021], Fedrecovery [Zhang *et al.*, 2023], gradient ascent-based: Amnesiac [Graves *et al.*, 2021], Pruning-based: Class-dis [Wang *et al.*, 2022] and the proposed FedAU to evaluate the effectiveness.

Evaluation Metrics. We performed backdoor detection and membership inference attack (MIA) [Gao *et al.*, 2022; Graves *et al.*, 2021] on the unlearned model to see if the influence of the targeted client was really removed by the proposed unlearning algorithm. The less the backdoor detection rate, and attack accuracy metric, the more effective the FMU methods are (Due to page limit, please refer the results of the recall in Appendix C). Moreover, the unlearning time cost and model performance of the remaining data is also utilized to evaluate all FMU methods. We refer all details about the experimental setting on Appendix A.

4.2 Overall Evaluation

Evaluation of Unlearning Effect

To ensure an effective unlearning method, it is crucial for the unlearned model to retain minimal information about the forgotten data. In Tab. 1, we present a comparison of accuracy of unlearning data achieved by various unlearning methods on the MNIST and CIFAR10 datasets. Our observations are as follows: 1) Amnesiac unlearning [Graves *et al.*, 2021] demonstrate strong performance in unlearning samples and clients, but exhibit lower effectiveness in unlearning classes (e.g., achieving a 20% increase in the accuracy of unlearning data compared to the retraining method); 2) Class-dis [Wang *et al.*, 2022] excel in unlearning classes, but are not suitable for unlearning samples and entire classes; 3) Our proposed method, FedAU, closely approximates the performance of retraining methods for unlearning samples, classes, and clients. For instance, the accuracy of unlearning data of FedAU is less than 1% compared to retraining methods.

Evaluation of Utility

In Table 1, we evaluate the utility of the remaining data by measuring the remaining accuracy. The results indicate: 1)

Dataset (%)	UL Method	FedAvg	Retraining		Amnesiac		Class-disc		FedEraser		FedAU	
		Rm-Acc	Ul/Rm-Acc		Ul/Rm-Acc		Ul/Rm-Acc		Ul/Rm-Acc		Ul/Rm-Acc	
CIFAR10 AlexNet	Samples	87.77 ± 0.21	1.90 ± 0.20	87.36 ± 0.23	4.8 ± 0.99	85.91 ± 0.14	—	—	—	—	0.35 ± 0.07	86.24 ± 0.16
	Classes	87.50 ± 0.05	0.00 ± 0.00	87.45 ± 0.28	26.15 ± 2.76	74.93 ± 3.38	0.00 ± 0.00	79.42 ± 1.25	—	—	0.01 ± 0.01	87.71 ± 0.33
	Clients	87.49 ± 0.10	1.80 ± 0.16	87.33 ± 0.19	6.05 ± 0.35	75.34 ± 1.44	—	—	9.32 ± 0.11	84.60 ± 0.74	0.52 ± 0.06	86.83 ± 0.31
MNIST LeNet	Samples	99.44 ± 0.02	0.44 ± 0.13	99.46 ± 0.04	1.65 ± 0.07	98.87 ± 0.21	—	—	—	—	0.62 ± 0.23	99.36 ± 0.01
	Classes	99.50 ± 0.04	0.00 ± 0.00	99.54 ± 0.02	53.78 ± 6.06	57.95 ± 3.15	0.00 ± 0.00	99.13 ± 0.16	—	—	0.00 ± 0.00	99.63 ± 0.01
	Clients	99.28 ± 0.06	0.77 ± 0.27	98.69 ± 0.04	0.53 ± 0.28	97.89 ± 0.47	—	—	8.05 ± 0.50	99.33 ± 0.21	0.66 ± 0.10	99.08 ± 0.12

Table 1: The comparison with current methods, including FedAvg, Retraining, Amnesiac unlearning [Graves *et al.*, 2021], Class-dis [Wang *et al.*, 2022] and Federaser [Liu *et al.*, 2021] and FedAU in different federated machine unlearning scenarios.

	Samples	Class	Client
Retraining	$\sim 10^3$	$\sim 10^3$	$\sim 10^3$
Amnesiac [Graves *et al.*, 2021]	$\sim 10^0$	$\sim 10^0$	$\sim 10^0$
FedEraser [Liu *et al.*, 2021]	/	/	$\sim 10^3$
Class-dis [Wang *et al.*, 2022]	/	$\sim 10^2$	/
FedRecovery [Zhang *et al.*, 2023]	/	$\sim 10^0$	/
FedAU (Ours)	$\sim 10^{-3}$	$\sim 10^{-3}$	$\sim 10^{-3}$

Table 2: Unlearning time cost (s) for different FMU methods under different federated machine unlearning scenarios.

The Amnesiac unlearning method [Graves *et al.*, 2021] and the Federaser method [Liu *et al.*, 2021] are both affected in terms of remaining accuracy. For instance, the drop in remaining accuracy for the unlearning class in CIFAR10 using the Amnesiac unlearning method is more than 10% compared to the retraining method; 2) Our proposed method, FedAU, effectively maintains the remaining accuracy with a minimal drop. Specifically, on CIFAR10, the drop in remaining accuracy is less than 1.5%, and on MNIST, it is only 0.2%.

Evaluation of Time Cost

Finally, we report the time consumed by each FMU (Fine-tuning Model Update) method to demonstrate the associated time costs (see details and more comparison on space consumption in Appendix C). The main results on the CIFAR10 dataset are presented in Tab. 2. From these results, we can draw two conclusions:

1. Among all the schemes, the Retraining scheme and schemes involving fine-tuning operations consume considerably more time compared to other methods;
2. Although the Amnesiac and FedRecovery scheme requires a relatively small amount of time for unlearning, they still several orders of magnitude slower than FedAU;
3. FedAU results in minimal additional training time, e.g, additional 2s for AlexNet-CIFAR10. This is because the AU module is lightweight such that training AU once consumes little time and training AU successfully only requires several epochs (see Appendix C).

4.3 Ablation Study

This section introduces the ablation study on the some important factors: the number of unlearning clients, the Non-IID extent and the coefficient (α, β) of FedAU. More ablation study on the proportion of unlearning samples, and impact of Coefficient α and β see in Appendix C.

Unlearning for Multiple Clients

In the scenario where multiple clients request to unlearn, we allocate each client to unlearn 10% of their respective datasets. The results in Figure 3 depict the accuracy of unlearning and remaining data as the number of unlearning clients varies. The graph demonstrates that as the number of unlearning clients increases, the accuracy of the unlearning and remaining datasets achieved by our proposed method, FedAU, approaches that of the retraining method. This observation highlights the generality and effectiveness of our method in the multiple client unlearning scenario.

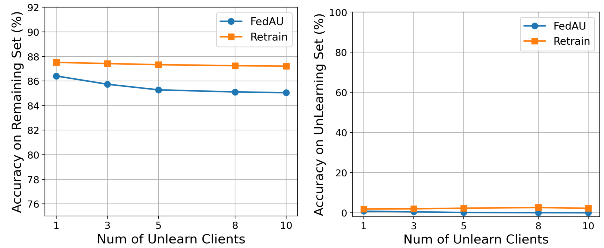


Figure 3: The accuracy of FedAU and retraining methods on CIFAR10 with different number of unlearning clients.

Impact of Non-IID Extent

In our study, we examined the impact of the Non-IID extent on the performance of the proposed FedAU (Federated Adaptive Unlearning) and retraining methods. To quantify the Non-IID extent, we used the $Dir(\gamma)$ distribution, where smaller values of γ indicate more heterogeneous data.

Fig. 4 illustrates the results of our experiments. We observed that the proposed FedAU method achieved a significant unlearning effect, as evidenced by the accuracy on the unlearning samples being less than 0.1% when $\gamma = 1$. Additionally, the FedAU method successfully maintained the model accuracy on the remaining data, with a drop of less than 2% compared to the retraining method.

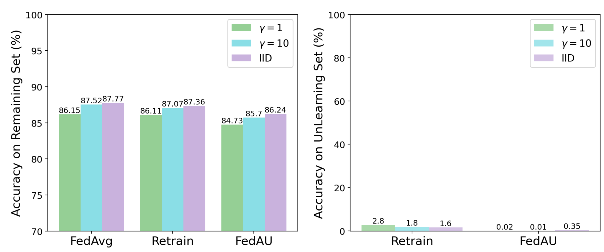


Figure 4: The impact of Non-IID on CIFAR10 for the proposed FedAU and Retraining methods.

5 Conclusion

In response to the limitations associated with unlearning in Federated Learning (FL), we have introduced FedAU, an innovative and efficient Federated Machine Unlearning (FMU) framework. Briefly, FedAU integrates a lightweight auxiliary unlearning module into the learning process, employing a straightforward linear operation to streamline unlearning without the need for additional time-consuming steps. Moreover, FedAU empowers multiple clients to simultaneously perform unlearning tasks and supports unlearning at various levels of granularity, ranging from individual data samples to specific classes and even client-level unlearning. We hope that its versatility and performance can make it a promising tool for future developments in the field.

Acknowledgements

This work is partly supported by National Natural Science Foundation of China (NO.62206154), Shenzhen Startup Funding (No.QD2023014C), and supported by the National Research Foundation, Singapore and Infocomm Media Development Authority under its Trust Tech Funding Initiative (No. DTC-RGC-04).

Contribution Statement

Hanlin Gu and Gongxi Zhu contributed equally to this work. Yuxing Han is the corresponding author.

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.

[Che *et al.*, 2023] Tianshi Che, Yang Zhou, Zijie Zhang, Lingjuan Lyu, Ji Liu, Da Yan, Dejing Dou, and Jun Huan. Fast federated machine unlearning with nonlinear functional theory. In *International conference on machine learning*, pages 4241–4268. PMLR, 2023.

[Chen *et al.*, 2023] Min Chen, Weizhuo Gao, Gaoyang Liu, Kai Peng, and Chen Wang. Boundary unlearning: Rapid forgetting of deep networks via shifting the decision boundary. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7766–7775, 2023.

[Cheng *et al.*, 2020] Yong Cheng, Yang Liu, Tianjian Chen, and Qiang Yang. Federated learning for privacy-preserving ai. *Communications of the ACM*, 63(12):33–36, 2020.

[Deng *et al.*, 2020] Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Distributionally robust federated averaging. *Advances in neural information processing systems*, 33:15111–15122, 2020.

[Dwork, 2006] Cynthia Dwork. Differential privacy. In *International colloquium on automata, languages, and programming*, pages 1–12. Springer, 2006.

[Eisenhofer *et al.*, 2022] Thorsten Eisenhofer, Doreen Riepel, Varun Chandrasekaran, Esha Ghosh, O. Ohrimenko, and Nicolas Papernot. Verifiable and provably secure machine unlearning. *ArXiv*, abs/2210.09126, 2022.

[Gao *et al.*, 2022] Xiangshan Gao, Xingjun Ma, Jingyi Wang, Youcheng Sun, Bo Li, Shouling Ji, Peng Cheng, and Jiming Chen. Verifi: Towards verifiable federated unlearning. *arXiv preprint arXiv:2205.12709*, 2022.

[Goldreich, 1998] Oded Goldreich. Secure multi-party computation. *Manuscript. Preliminary version*, 78(110):1–108, 1998.

[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.

[Halimi *et al.*, 2022] Anisa Halimi, Swanand Ravindra Kadhe, Ambrish Rawat, and Nathalie Baracaldo Angel. Federated unlearning: How to efficiently erase a client in fl? In *International Conference on Machine Learning*, 2022.

[Harding *et al.*, 2019] Elizabeth Liz Harding, Jarno J Vanto, Reece Clark, L Hannah Ji, and Sara C Ainsworth. Understanding the scope and impact of the california consumer privacy act of 2018. *Journal of Data Protection & Privacy*, 2(3):234–253, 2019.

[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.

[He *et al.*, 2019] Zecheng He, Tianwei Zhang, and Ruby B Lee. Model inversion attacks against collaborative inference. In *Proceedings of the 35th Annual Computer Security Applications Conference*, pages 148–162, 2019.

[Jose and Simeone, 2021] Sharu Theresa Jose and O. Simeone. A unified pac-bayesian framework for machine unlearning via information risk minimization. *2021 IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, 2021.

[Konečnỳ *et al.*, 2015] Jakub Konečnỳ, Brendan McMahan, and Daniel Ramage. Federated optimization: Distributed optimization beyond the datacenter. *arXiv preprint arXiv:1511.03575*, 2015.

[Krizhevsky *et al.*, 2012] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[Krizhevsky *et al.*, 2014] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. The cifar-10 dataset. *online: http://www.cs. toronto. edu/kriz/cifar. html*, 55(5), 2014.

[LeCun *et al.*, 1998] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[LeCun *et al.*, 2010] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

[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.

[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.*, 2022a] 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.*, 2022b] 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. *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications*, pages 1749–1758, 2022.

[Liu *et al.*, 2023] Ziyao Liu, Yu Jiang, Jiyuan Shen, Minyi Peng, Kwok-Yan Lam, and Xingliang Yuan. A survey on federated unlearning: Challenges, methods, and future directions. *arXiv preprint arXiv:2310.20448*, 2023.

[McMahan *et al.*, 2017] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017.

[Mercuri *et al.*, 2022] Salvatore Mercuri, Raad Khraishi, Ramin Okhrati, Devesh Batra, Conor Hamill, Taha Ghasempour, and Andrew Nowlan. An introduction to machine unlearning. *arXiv preprint arXiv:2209.00939*, 2022.

[Nasr *et al.*, 2019] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE symposium on security and privacy (SP)*, pages 739–753. IEEE, 2019.

[Sekhari *et al.*, 2021] Ayush Sekhari, Jayadev Acharya, Gautam Kamath, and Ananda Theertha Suresh. Remember what you want to forget: Algorithms for machine unlearning. *Advances in Neural Information Processing Systems*, 34:18075–18086, 2021.

[Su and Li, 2023] Ningxin Su and Baochun Li. Asynchronous federated unlearning. In *IEEE INFOCOM 2023-IEEE Conference on Computer Communications*, pages 1–10. IEEE, 2023.

[Sun *et al.*, 2022] Tao Sun, Dongsheng Li, and Bao Wang. Decentralized federated averaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4289–4301, 2022.

[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.

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

[Yang *et al.*, 2019] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

[Zhang *et al.*, 2023] Lefeng Zhang, Tianqing Zhu, Haibin Zhang, Ping Xiong, and Wanlei Zhou. Fedrecovery: Differentially private machine unlearning for federated learning frameworks. *IEEE Transactions on Information Forensics and Security*, 2023.