

Federated Unlearning in the Wild: Rethinking Fairness and Data Discrepancy

ZiHeng Huang^{1*}, Di Wu^{2*}, Jun Bai^{3†}, Jiale Zhang⁴, Sicong Cao⁵, Ji Zhang⁶, Yingjie Hu¹

¹ Central South University

² La Trobe University

³ McGill University

⁴ Yangzhou University

⁵ Nanjing University of Posts and Telecommunications

⁶ University of Southern Queensland

Abstract

Machine unlearning has become a critical capability to support data deletion rights, such as the “right to be forgotten” mandated by privacy regulations. As a decentralized learning paradigm, Federated Learning (FL) also faces growing demands for unlearning. However, enabling unlearning in realistic FL settings presents two major challenges. First, fairness in FU is often overlooked: 1) Existing exact unlearning technical Routes typically require all clients to participate in retraining, regardless of their involvement in the unlearning request, leading to unnecessary computation and communication overhead; 2) recent approximate approaches apply gradient ascent, distillation or directly zero out neurons associated with the forget set, but such coarse interventions neglect their relevance to retained knowledge. This can unfairly degrade performance for clients whose data is entirely from the retain set. Second, data distribution discrepancy poses a significant challenge. Most existing evaluations rely on artificially synthetic IID or non-IID assumptions that fail to reflect the natural heterogeneity of real-world federated systems. These unrealistic benchmarks obscure the true impact of unlearning on both local and global utility and limit the applicability of current methods in production environments. To bridge this gap, we conduct a comprehensive benchmark of existing FU technical Routes under both fairness and realistic data heterogeneity conditions. Furthermore, we propose a novel and fairness-aware FU approach, namely Federated Cross-Client-Constrains Unlearning (FedCCCU), that explicitly addresses both challenges. FedCCCU offers a practical and scalable solution for real-world FU, providing a foundation for future research in this area. Experimental results show that existing methods perform poorly under realistic data settings, while our approach consistently outperforms them across diverse, real-world scenarios.

Introduction

With the widespread adoption of machine learning, the need for user data deletion and compliance with privacy regulations has become increasingly critical. This demand is further reinforced by legal frameworks such as the GDPR (Regulation, Protection 2018) and CCPA (Goldman 2020), both

of which explicitly grant users the right to request data deletion. In response, Machine Unlearning (MU) (Bourtoule et al. 2021; Li et al. 2025) has emerged as a key technique for realizing the “right to be forgotten,” achieving promising results in centralized settings. Common strategies involve retraining or targeted model interventions to remove the influence of specific data from learned parameters. However, these centralized approaches are not directly applicable to Federated Learning (FL), where a global model is formed by aggregating locally trained updates from distributed clients. The collaborative and decentralized nature of FL fundamentally differs from the centralized paradigm, making retraining-based unlearning infeasible at the individual client level. Consequently, there is an urgent need to develop FU methods (Liu et al. 2024) that can ensure regulatory compliance while maintaining strong privacy guarantees in distributed environments.

Current federated unlearning research primarily follows two mainstream technical routes: exact unlearning and approximate unlearning. The exact unlearning route aims to restore the model to a state as if the target data had never been involved in training. This is typically achieved through strategies such as delete-and-retrain or label relabeling, which require global retraining involving all clients. Representative methods include FedEraser (Liu et al. 2021a), VeriFi (Gao et al. 2024), SFU (Li et al. 2023), and KNOT (Su and Li 2023). These methods typically require all clients to participate in global retraining, resulting in substantial computational and coordination overhead. The approximate unlearning route seeks to balance unlearning effectiveness with practicality by avoiding full retraining. It encompasses a variety of techniques such as knowledge distillation, gradient editing, and model editing, which aim to suppress or erase the influence of specific data from the model. Representative methods in this category include MoDe (Zhao et al. 2023), GA (Jang et al. 2022; Halimi et al. 2022), FedFilter (Wang et al. 2023), 2F2L (Jin et al. 2023), FedRecovery (Cao et al. 2023), and DEPNN (Wu et al. 2023). These approaches generally perform well in small-scale settings or under synthetically partitioned non-IID data, and currently represent the dominant direction of research in FU.

However, existing paradigms rely on two fragile assump-

*These authors contributed equally.

†Corresponding author: baijun@deakin.edu.au

tions. First, current exact unlearning methods enforce global retraining across all clients regardless of their involvement, and approximate unlearning that overlooks the importance of retained knowledge, which together raise fairness concerns at both the system and model levels. Second, evaluations typically simulate non-IID conditions via label-based partitioning, which fails to reflect real-world cross-domain feature heterogeneity and thus overstates the robustness of current methods.

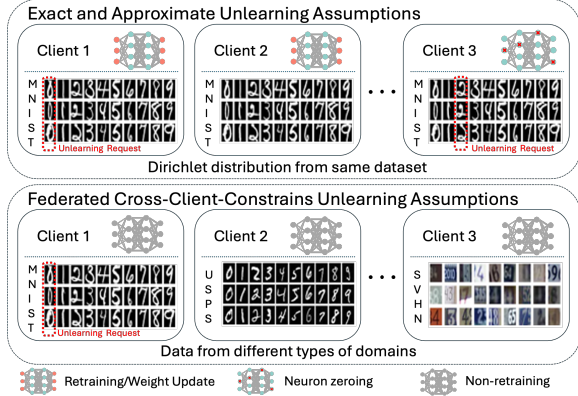


Figure 1: Federated unlearning in single-domain (top) and cross-domain (bottom) settings.

Figure 1 illustrates the gap between conventional and realistic FU settings. Prior works often simulate non-IID data via Dirichlet-based label splits over a single dataset (“pseudo-Noniid”), ignoring real-world scenarios where clients from different domains (e.g., schools, banks, postal services) share label spaces but differ in feature distributions. Existing FU technical Routes typically enforce global retraining or model editing across all clients, assuming full cooperation and fairness, yet they either impose redundant costs on uninvolved clients or degrade unrelated knowledge through coarse neuron removal. In contrast, our Federated Cross-Client-Constrained Unlearning (FedCCCU) addresses both issues by considering cross-domain heterogeneity and constraining model editing to minimize collateral impact on non-forgetting clients, making FU fairer and more practical. The contributions of this paper can be summarized below.

- We formalize two overlooked challenges in federated unlearning: fairness and data distribution discrepancy in the cross-domain settings for the different technical routes.
- We conduct an extensive evaluation of representative exact and approximate FU methods under our benchmark and reveal their limitations in both forgetting effectiveness and fairness in realistic federated learning scenarios.
- We propose a novel fairness-aware unlearning method (FedCCCU) and advocate for future research to prioritize fairness and data realism alongside accuracy, paving the way for deployable and responsible federated unlearning systems.

Related Work

Existing FU research still follows a “retrain-all plus synthetic data” paradigm, and its evolution traces several intersecting lines. FedEraser(Liu et al. 2021a) cancels historical gradients by backtracking to replicate a full retrain, but the computational and costs are spread across every client, creating fairness issues. Subsequent work tried to lighten this load: Ferrari(Gu et al. 2024) deletes sensitive features in embedding space; (Lin et al. 2024) shard clients and encode within each shard to accelerate retraining; Deepoblivate(He et al. 2021) quantizes residual memories and trims iterations online; (Liu et al. 2022) use a first-order Taylor approximation of the loss to finish retraining in just a few rounds; and SIFU(Fraboni et al. 2024) combines time rollback with sequential re-optimization, clipping rollback points via bounded sensitivity before incrementally updating the remaining data.

Although these strategies improve efficiency, they all assume that CIFAR10/100 “pseudo-noniid” slices adequately represent real, cross-island distributions. To reduce overhead further, researchers have proposed approximate unlearning techniques. Knowledge distillation and soft-label recovery ((Wu, Zhu, and Mitra 2022), MoDe(Zhao et al. 2023), (Zhu, Li, and Hu 2023)) fade memory by relaxing equivalence; gradient ascent and feature filtering (GA(Jang et al. 2022)(Gu et al. 2024), FedFilter(Wang et al. 2023)) directly intervened in gradient directions; pruning methods (RevFRF(Liu et al. 2021b), (Wang et al. 2022)) excise class-specific weights and then fine-tune; FedRecovery(Cao et al. 2023) injected differential privacy noise into residual gradients, while Depn(Wu et al. 2023) provided millisecond-level responses through targeted weight pruning. VeriFi(Gao et al. 2024) augments participation-heavy frameworks with zero-knowledge proofs, enhancing auditability but further increasing the resource burden imposed by global retraining. In sum, current FU techniques perform well on small, homogeneous datasets but remain untested in truly heterogeneous, real-world environments.

In contrast to unlearning tasks, the mainstream FL community has extensively explored fairness in resource allocation and performance, exemplified by personalized alignment approaches (e.g., FedDyn(Acar et al. 2021), FedBN(Li et al. 2021)). Concurrently, robust optimization methods addressing data heterogeneity (Karimireddy(Karimireddy, He, and Jaggi 2021)) and realistic non-iid benchmarks (such as Leaf(Caldas et al. 2018) and CORA(McCallum et al. 2000)) have become increasingly mature. However, these advances have not yet been transferred to the unlearning domain.

The abovementioned FU methods lack comprehensive consideration of realistic data distributions and unlearning deployment patterns, as summarized in Table 1. Most existing FU approaches are still confined to synthetic IID or pseudo-noniid splits, and their retraining strategies fall into only two categories: (i) full-scale retraining that involves every client, or (ii) partial retraining in which, beyond the forgetting client, a subset of additional clients also participates.

Method	Year	Multi-client Retraining	Data Distribution
FedEraser	2021	✓	IID / pseudo-noniid
Deepobliviate	2021	✓	pseudo-noniid
Revfrf	2021	✓	pseudo-noniid
Wu et al	2022	✓	pseudo-noniid
Liu et al	2022	✓	pseudo-noniid
Wang et al	2022	✓	pseudo-noniid
MoDe	2023	×	pseudo-noniid
FedFilter	2023	✓	pseudo-noniid
FedRecovery	2023	×	pseudo-noniid
KNOT	2023	✓	IID / pseudo-noniid
FedLU	2023	✓	pseudo-noniid
Ferrari	2024	×	IID
Lin et al	2024	✓	IID / pseudo-noniid
VeriFi	2024	×	pseudo-noniid
SIFU	2024	✓	pseudo-noniid
FedCCCU (Ours)	–	×	real-noniid

Table 1: The assumptions of different FU methods. “Multi-client Retraining” indicates whether retraining involves clients beyond the one requesting unlearning. “Pseudo-noniid” splits a single dataset across clients, while “Real-noniid” assigns distinct datasets to distinct clients.

Problem Formulation

Technical Routes

- **Delete-Retrain:** Considered the “gold standard” for exact unlearning, this route removes all sensitive samples from the forgetting client’s local dataset and retrains the model globally.
- **Relabel-Poison:** this route avoids deleting data. Instead, it randomly rewrites the labels of class-0 samples to other non-target classes, gradually weakening the model’s original decision boundary.
- **Neuron-Zeroing:** this route first performs a sensitivity analysis on the global model to identify the most activated neurons or channels for class-0 during forward propagation. These parameters are then zeroed out.

Fair Retraining Dilemma

In a typical federated learning system, a central server collaborates with K clients to minimize the global risk, formally defined as:

$$F(\mathbf{w}) = \sum_{k=1}^K \frac{n_k}{n} \mathbb{E}_{(x,y) \sim \mathcal{D}_k} [\ell(\mathbf{w}; \mathbf{x}, y)] \quad (1)$$

where $\mathbf{w}$ denotes the model parameters, $\mathcal{D}_k$ represents the local dataset of client k containing n_k samples, and $n = \sum_{k=1}^K n_k$. The global loss is computed as the weighted aggregation of each client’s expected loss E_k , proportional to their local data sizes. Training proceeds until the validation error falls below a predefined threshold ε , and the communication round at which this first occurs is recorded as the convergence round T_0 .

If a subset of clients C_{req} later requests data removal, existing methods require retraining across all clients. This imposes unnecessary computational and communication overhead on non-requesting clients, raising fairness concerns.

The Pseudo-Noniid Fallacy

Existing studies typically partition a single dataset D among multiple clients via Dirichlet sampling. Although this approach modifies each client’s label priors $p_k(y)$, it implicitly assumes all clients share an identical conditional distribution. However, Real-Noniid scenario often differ significantly: even when tasks and labels uniformly involve , data distributions across clients, such as chalkboard photos, touchscreen signatures, and scanned images, vary substantially due to differing imaging processes. It means $p_k(x | y) \neq p_j(x | y)$.

Cross-Domain FU Benchmark Across Technical Routes

To benchmark the performance of the existing technical route on cross-domain FU. We propose Cross Domain FU, a benchmarking framework designed for realistic federated unlearning. Unlike prior settings that partition a single dataset, it treats each client as an autonomous data silo with heterogeneous features and labels. Upon an unlearning request, only the requesting clients retrain locally, while others retain previous model states. The server then aggregates updates until convergence. This protocol, based on local retraining and global inheritance, better reflects real-world deployment by aligning both data distribution and training dynamics with practical constraints.

Dataset Overview and Selection Rationale

In constructing the Cross Domain FU Benchmark, we adhere to a core principle: ensuring that all clients perform an identical classification task while maximizing divergence in their local feature distributions.

We select six widely used datasets, namely MNIST10 (LeCun et al. 1998), SVHN (Netzer et al. 2011), USPS (Hull 1994), CIFAR10 (Krizhevsky and Hinton 2009), CIFAR100 (Krizhevsky and Hinton 2009), and ImageNet (Russakovsky et al. 2015), whose basic statistics are summarized in Table 2. To ensure label consistency across domains, we retain only the overlapping classes, resulting in 9 shared labels between CIFAR10 and ImageNet, and 65 between CIFAR100 and ImageNet. These dataset pairs maintain a unified label space while presenting significant visual disparities. For instance, SVHN contains cluttered RGB backgrounds, whereas MNIST consists of clean binary images; similarly, ImageNet features rich textures, in contrast to the compressed visual patterns of CIFAR. This cross-domain variation offers a realistic foundation for evaluating federated unlearning under heterogeneous feature distributions.

Data distribution strategy

To better address the “data realism gap,” we design a Real-Noniid partitioning strategy that maintains task consistency while inducing realistic feature heterogeneity across clients.

Task	Dataset	Number of Classes	Training Samples	Test Samples
Handwritten Digit Recognition	MNIST10	10	60000	10000
	SVHN	10	73257	26032
	USPS	10	7291	2007
Image Recognition	CIFAR10	10	50000	10000
	CIFAR100	100	50000	10000
	ImageNet	1000	1280000	50000

Table 2: Overview of the Dataset

Client-wise data allocations under each strategy are detailed in Table 3.

In the task, we assign MNIST10, SVHN, and USPS to clients C1–C3, C4–C6, and C7–C9 respectively, with intra-group samples partitioned via a Dirichlet distribution. Each client thus receives data from a distinct source domain.

For image recognition, we construct two intersection-based scenarios: one with 9 shared classes from CIFAR10 and ImageNet, and another with 65 shared classes from CIFAR100 and ImageNet. In both cases, we allocate low-resolution CIFAR images (32×32) to five clients and high-resolution ImageNet images (224×224) to the other five.

Split Scenario	Client	Dataset / Range	Resolution
Real-Noniid	C1-C3	MNIST10(0–9)	28×28
	C3-C6	SVHN(0–9)	32×32
	C7-C9	USPS(0–9)	16×16
	C1-C5	CIFAR10(0–8)	32×32
	C6-C10	ImageNet(0-8)	224×224
	C1-C5	CIFAR100(0–64)	32×32
	C6-C10	ImageNet(0-64)	224×224

Table 3: Client-wise data allocation results under different split scenarios

This ‘task-aligned but domain-divergent’ setup ensures that Real-Noniid maintains label comparability while introducing multi-dimensional distribution shifts across factors such as resolution, illumination, texture, and capture modality, which are frequently encountered in real-world cross-organization federated systems.

To visualize the differences between our Real-Noniid partition and existing data distribution assumptions, Figure 2 illustrates a 3D bar chart, where the x-axis denotes client IDs, the y-axis indicates the number of classes per client, and the z-axis represents average image resolution as a proxy for feature heterogeneity. In the IID setting, all bars are uniform in both height and depth. In Pseudo-Noniid, class distributions vary across clients, but resolution remains constant. In contrast, Real-Noniid maintains equal label coverage while exhibiting significant differences in resolution, highlighting the most realistic and challenging distribution scenario.

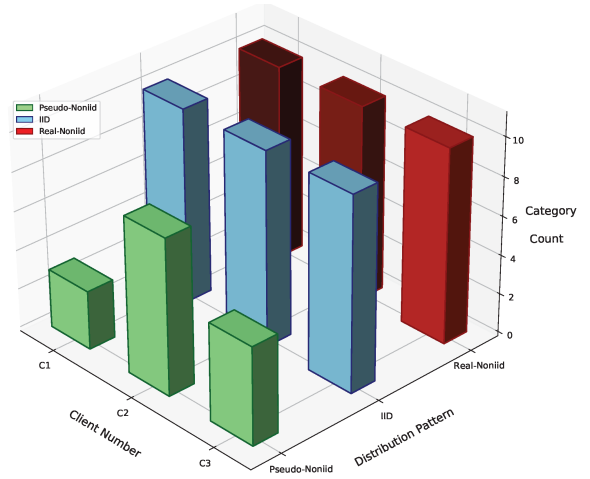


Figure 2: The distribution of data feature characteristics of the client in different allocation scenarios

Benchmark Experiments

Previous studies have confirmed the effectiveness of the existing federal forgetting learning techniques in IID and pseudo-Noniid scenarios, Table 4 reports results under the *Real-Noniid* setting. Under the Real-Noniid setting, mainstream federated unlearning technical routes, including exact retraining and approximate model editing, show clear limitations. In , exact retraining fails to achieve effective forgetting, with target class accuracy remaining above 96%, while approximate editing reduces it to 0% but causes a 7% drop in overall accuracy and degrades the model’s accuracy on remaining classes. In image recognition, exact retraining yields only partial forgetting (a 10–17% drop) with performance loss, while approximate editing is more aggressive (e.g., from 94.33% to 7.94% or from 60.31% to 0%) but severely harms other classes and clients.

Table 5 and Table 6 shows the accuracy variations of different clients on label_0 data before and after applying federated unlearning in various tasks. By comparing the effects of different unlearning strategies, we observe the following key points.

In the task, mainstream precise unlearning techniques failed to significantly reduce label_0 accuracy on the forgotten clients due to high inter-client data homogeneity. In contrast, approximate techniques such as Neuron-Zeroing reduced the accuracy to 0% across all clients, including those not involved in the unlearning request, reflecting a typical case of over-forgetting. In the image recognition task, both mainstream unlearning technical routes led to limited forgetting on the target client. Specifically, on client0, label_0 accuracy only decreased from 94.33% to 82.77% and from 62.86% to 45.71%, indicating a partially effective but insufficient forgetting outcome. Meanwhile, performance on non-target clients also deteriorated, with label_0 accuracy on client9 dropping by 26%. This suggests that local unlearning perturbations may propagate through model aggregation, resulting in unfair cross-client degradation.

Evaluation Results under Real-Noniid Setting

Data1	Data2	Data3	Model	Global Accuracy				Client C1 (Class-0) Accuracy			
				Original	Delete	Relabel	Zeroing	Original	Delete	Relabel	Zeroing
MNIST10	SVHN	USPS	CNN	94.09%	95.70%	95.59%	87.16%	98.95%	97.90%	96.50%	0.00%
CIFAR10	ImageNet	–	ResNet18	90.88%	88.46%	88.62%	73.66%	94.33%	82.77%	83.67%	7.94%
–	ImageNet	CIFAR100	ResNet32	69.46%	72.99%	69.90%	60.31%	62.86%	45.71%	51.43%	0%

Table 4: Evaluation results under the Real-Noniid setting. Only the client issuing the forgetting request is retrained; all others retain their pretrained global model. Delete-Retrain, Relabel-Poison, and Neuron-Zeroing represent three unlearning strategies.

Task	Client	Original	Delete	Relabel	Zeroing
Handwriting Digit Recognition	client1	98.77%	97.90%	96.50%	0.00%
	client2	99.36%	98.08%	95.83%	0.00%
	client3	98.97%	97.94%	98.97%	0.00%
	client4	96.35%	97.08%	96.35%	0.00%
	client5	95.72%	96.20%	94.06%	0.00%
	client6	96.99%	96.99%	94.64%	0.00%
	client7	28.57%	53.57%	89.29%	0.00%
	client8	26.13%	52.26%	90.24%	0.00%
	client9	27.27%	52.27%	90.91%	0.00%

Table 5: Per-client Label_0 accuracy for the Handwriting Digit Recognition task before and after unlearning.

Nevertheless, in some specific cases, unlearning strategies may have positive effects. For example, in the Image Recognition(65) task, the accuracy of client5 and client8 improved from 60% and 33.33% to 73.33% and 66.67%, respectively. This phenomenon indicates that the unlearning process not only removes irrelevant or conflicting gradients but can also serve as a regularization technique, improving model performance on heterogeneous clients.

In summary, existing FU strategies exhibit significant variability in their effects in real-world applications, particularly when data distribution heterogeneity is high. The performance of precise and approximate unlearning strategies across different tasks reveals both positive and negative effects that may arise when models face multi-client data heterogeneity.

Proposed Method

To address the critical deficiencies of existing FU methods in terms of fairness and the modeling of data distributions, we propose a novel FU method named Federated Cross-Client-Constrains Unlearning(FedCCCU). This method is specifically designed for realistic cross-domain data distribution scenarios, aiming to achieve effective unlearning while minimizing the performance impact on non-requesting clients. FedCCCU introduces a cross-client constraint mechanism, which, combined with a lightweight model editing strategy, enhances the method’s deployability and robustness. Figure 3 illustrates the overall workflow of FedCCCU.

Identification of Key Neurons

Our method for identifying key neurons associated with specific classes is primarily inspired by the DEPN(Wu et al.

Task	Client	Original	Delete	Relabel	Zeroing
Image Recognition (9)	client1	94.33%	82.77%	83.67%	7.94%
	client2	100.00%	77.78%	77.78%	5.56%
	client3	94.05%	83.78%	84.86%	9.73%
	client4	94.59%	83.78%	85.41%	8.65%
	client5	97.08%	85.96%	83.04%	8.19%
	client6	90.32%	83.87%	80.06%	17.01%
	client7	92.41%	79.11%	83.54%	17.72%
	client8	90.74%	77.78%	85.19%	14.81%
	client9	100.00%	73.91%	73.91%	17.39%
	client10	93.55%	79.03%	84.68%	16.94%
Image Recognition (65)	client1	62.86%	45.71%	51.43%	0.00%
	client2	57.14%	57.14%	71.43%	0.00%
	client3	50.00%	50.00%	50.00%	0.00%
	client4	53.85%	38.46%	53.85%	0.00%
	client5	76.74%	62.79%	69.77%	0.00%
	client6	60.00%	73.33%	73.33%	0.00%
	client7	75.00%	50.00%	75.00%	0.00%
	client8	80.70%	78.95%	78.95%	0.00%
	client9	33.33%	66.67%	66.67%	0.00%
	client10	71.43%	61.90%	61.90%	0.00%

Table 6: Per-client Label_0 accuracy for Image Recognition tasks before and after unlearning. Image Recognition(9/65) denotes tasks with 9 and 65 classes.

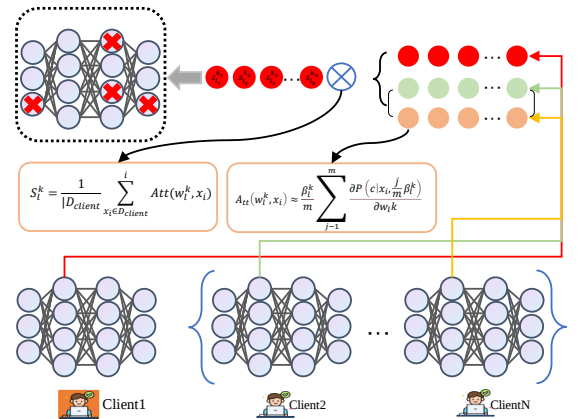


Figure 3: FedCCCU: Federated Cross-Client-Constrains Unlearning

2023) framework. DEPN effectively quantifies the contribution of individual neurons to the model’s output through a gradient-based attribution approach. We apply this concept within our federated learning framework, aiming to identify sensitive neurons in the model that are highly associated with the class that needs to be forgotten (e.g., label=0).

The entire identification process is executed locally at each client within the federated learning framework, ensuring the privacy of client data is preserved. The specific procedure is as follows: When the server receives a forgetting request from a client, each client will receive the current global model parameters, denoted as θ . The client’s objective is to compute the sensitivity score for each neuron in the model, for every class, based on the data set D_{client} it holds locally.

For any neuron w in the model (where l represents the layer index and k is the neuron index within that layer), its contribution to classifying a single data sample x_i as the target class c can be quantified by calculating the cumulative gradient of the class prediction probability $P(c|x_i, \theta)$ as its activation value changes from 0 to its original value β . This contribution, referred to as the Attribution Score, is computed as follows:

$$\text{Att}(w_l^k, x_i) = \beta_l^k \int_0^{\beta_l^k} \frac{\partial P(c|x_i, \alpha_l^k)}{\partial w_l^k} d\alpha_l^k \quad (2)$$

- β_l^k is the original activation value of neuron w_l^k when the input is x_i .
- $P(c|x_i, \alpha_l^k)$ represents the probability that the model predicts input x_i as class c when the activation value of neuron w_l^k is temporarily set to α_l^k .
- $\frac{\partial P(\cdot)}{\partial w_l^k}$ is the partial derivative of the class prediction probability with respect to the neuron w_l^k , i.e., the gradient.

Given that directly computing the continuous integral is difficult, we approximate it using the Riemann sum, discretizing the integration process into m steps:

$$\text{Att}(w_l^k, x_i) \approx \frac{\beta_l^k}{m} \sum_{j=1}^m \frac{\partial P(c|x_i, \frac{j}{m}\beta_l^k)}{\partial w_l^k} \quad (3)$$

- $\frac{j}{m}\beta_l^k$ represents the activation value of the neuron at the j -th discrete step.

The client calculates attribution scores for all neurons for each sample in its local dataset D_{client} . Subsequently, by averaging the attribution scores of all samples in the dataset, a final sensitivity score S_l^k is obtained for each neuron w_l^k on that client.

$$S_l^k = \frac{1}{|D_{client}|} \sum_{x_i \in D_{client}} \text{Att}(w_l^k, x_i) \quad (4)$$

- $|D_{client}|$ is the total number of samples in the client’s local dataset D_{client} .

For a given class c , a neuron’s score indicates the strength of its association with that class, with higher scores reflecting stronger associations. In the end, each client will upload

the indices (l, k) of the topN most sensitive neurons, along with their corresponding sensitivity scores S , for each class to the central server.

Neuron Dominant Computing

In this section, we propose a method to measure the importance of neurons in the forgetting and non-forgetting clients. Based on this new idea, we define the concept of ”dominant neurons.” Specifically, we aim to assess the importance of each neuron for the forgetting client and its importance for all non-forgetting clients.

First, we select a list of sensitive neurons from all clients that belong to the same class as the forgetting data category and iterate through each neuron N . For each neuron N , we calculate its contribution to the forgetting client (Client 0), denoted as S_{forget} . Next, we search for the list of sensitive neurons that belong to the same class as this neuron N in all non-forgetting clients (Client 1, Client 2, ...), and find the maximum contribution of neuron N in each non-forgetting client, denoted as $S_{\text{max_other}}$. If N does not appear in the list of any non-forgetting client, then $S_{\text{max_other}} = 0$.

Then, we calculate the ratio R :

$$R = \frac{S_{\text{max_other}}}{S_{\text{forget}}} \quad (5)$$

Based on the value of ratio R , we define the importance of a neuron as follows:

- If R is large, it indicates that the neuron is crucial for some non-forgetting clients and should not be modified arbitrarily.
- If R is close to 1, it means the neuron has similar importance in both the forgetting and non-forgetting clients, and is referred to as a ”shared neuron.”
- If R is small, it indicates that the contribution of the neuron to the forgetting client is significantly greater than its contribution to any non-forgetting client, and it can be considered a ”dominant neuron.”

Based on this theory, we propose the strategy of ”ranking neurons by dominant score from low to high and selecting the first n neurons,” called ”Rank-Based Selection.” We will then edit the model by using the indices (l, w) of the selected top n neurons and set the corresponding weights of these neurons to zero.

Experimental Analysis

After modifying the model, we selected more complex Image Recognition tasks (Image Recognition9 and Image Recognition65) for experimental analysis. The dataset was then partitioned using the Real-Noniid method, and the experiments were conducted following the principle of training fairness. Figure 4 presents the overall accuracy changes before and after unlearning. We can observe that the overall accuracy drop for the Delete-Retrain and Relabel-Poison techniques is quite limited after unlearning. Next is our proposed method, and finally, the zeroing-out approximate unlearning technique, which causes a significant decrease in overall accuracy.

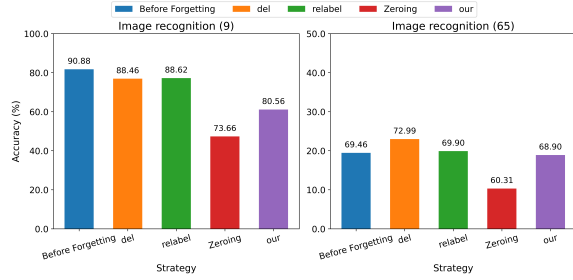


Figure 4: Performance Comparison of Different Unlearning Strategies on an Image Recognition Task

To provide a more comprehensive analysis, Table 7 and Table 8 show the accuracy performance of the top three clients across different categories. This table details the accuracy on both the class targeted for unlearning (the forgotten class) and all remaining classes. This enables a more detailed observation of the effects of different unlearning methods across various categories. For the 65-category Image Recognition task, due to the large number of classes, we only present the accuracy of the top 10 categories for the first three clients to provide a clearer view of the results.

Client	Label	Before	Delete	Relabel	Zeroing	Our
Client 1	label_0(↓)	94.33%	82.77%	83.67%	7.94%	16.55%
	label_1	96.30%	96.54%	97.28%	92.35%	94.57%
	label_2	90.89%	90.65%	93.29%	79.86%	89.69%
	label_3	88.78%	89.28%	85.04%	89.28%	90.52%
	label_4	91.14%	87.95%	92.50%	72.73%	87.27%
	label_5	95.95%	96.19%	96.67%	77.62%	95.48%
	label_6	95.58%	95.09%	95.82%	80.59%	93.12%
	label_7	97.06%	96.08%	97.79%	83.33%	95.10%
	label_8	96.22%	96.89%	96.00%	87.56%	95.78%
Client 2	label_0	100.00%	77.78%	77.78%	5.56%	27.78%
	label_1	88.89%	94.44%	100.00%	83.33%	88.89%
	label_2	88.46%	84.62%	84.62%	84.62%	88.46%
	label_3	91.30%	86.96%	91.30%	100.00%	86.96%
	label_4	86.36%	81.82%	81.82%	59.09%	86.36%
	label_5	90.91%	86.36%	90.91%	72.73%	90.91%
	label_6	100.00%	95.24%	95.24%	90.48%	95.24%
	label_7	100.00%	100.00%	100.00%	95.00%	100.00%
	label_8	84.62%	92.31%	84.62%	76.92%	84.62%
Client 3	label_0	94.05%	83.78%	84.86%	9.73%	20.54%
	label_1	97.03%	97.03%	97.03%	91.09%	95.05%
	label_2	92.75%	91.19%	92.23%	76.68%	91.71%
	label_3	89.42%	88.46%	87.02%	87.50%	88.46%
	label_4	92.31%	92.31%	91.72%	78.70%	89.35%
	label_5	97.79%	97.24%	97.79%	79.01%	97.24%
	label_6	95.63%	98.06%	97.57%	76.70%	96.12%
	label_7	97.21%	96.09%	97.21%	86.03%	97.21%
	label_8	96.92%	98.46%	96.41%	92.31%	96.41%

Table 7: Comparison of different unlearning methods on the Image Recognition(9) task across three clients.

From Table 7, we observe that while Delete-Retrain and Relabel-Poison result in only minor degradation in overall model performance, their forgetting efficacy remains limited. In contrast, the Zeroing route achieves stronger forgetting on the target client (e.g., class 0 accuracy on client1

drops from 94.33% to 7.94%), but introduces severe collateral effects. For example, class 0 accuracy on client2 and client3 drops to 5.56% and 9.73%, respectively, and class 4 on client2 decreases by 27.27%.

In comparison, our method reduces class 0 accuracy on client1 to 16.55%, achieving a comparable forgetting effect while substantially mitigating side effects. The accuracy of non-forgotten classes and clients remains largely unaffected. For instance, class 6 on client2 drops by only 4.76%. These results demonstrate the effectiveness of our approach in balancing forgetting performance and cross-client stability.

Client	Label	Before	Delete	Relabel	Zeroing	Our
Client 1	label_0(↓)	50.00%	50.00%	50.00%	0.00%	50.00%
	label_1	100.00%	100.00%	100.00%	100.00%	100.00%
	label_2	50.00%	100.00%	100.00%	50.00%	50.00%
	label_3	100.00%	100.00%	100.00%	100.00%	100.00%
	label_4	50.00%	50.00%	0.00%	0.00%	50.00%
	label_5	100.00%	100.00%	100.00%	100.00%	100.00%
	label_6	33.33%	33.33%	33.33%	33.33%	33.33%
	label_7	100.00%	100.00%	100.00%	0.00%	100.00%
	label_8	0.00%	0.00%	0.00%	0.00%	0.00%
Client 2	label_9	33.33%	33.33%	0.00%	0.00%	33.33%
	label_0	53.85%	38.46%	53.85%	0.00%	53.85%
	label_1	62.50%	62.50%	68.75%	56.25%	56.25%
	label_2	25.00%	50.00%	25.00%	12.50%	25.00%
	label_3	81.82%	90.91%	81.82%	72.73%	72.73%
	label_4	63.64%	81.82%	63.64%	54.55%	63.64%
	label_5	100.00%	100.00%	100.00%	100.00%	100.00%
	label_6	69.23%	38.46%	53.85%	53.85%	69.23%
	label_7	87.50%	87.50%	87.50%	62.50%	87.50%
Client 3	label_8	68.75%	81.25%	75.00%	62.50%	68.75%
	label_9	44.44%	33.33%	55.56%	44.44%	44.44%
	label_0	76.74%	62.79%	69.77%	0.00%	55.81%
	label_1	73.17%	68.29%	63.41%	63.41%	70.73%
	label_2	30.95%	35.71%	30.95%	19.05%	28.57%
	label_3	75.56%	88.89%	64.44%	53.33%	68.89%
	label_4	61.54%	69.23%	56.41%	43.59%	61.54%
	label_5	88.24%	88.24%	85.29%	85.29%	88.24%
	label_6	65.79%	55.26%	65.79%	60.53%	63.16%
	label_7	63.89%	86.11%	80.56%	47.22%	66.67%
	label_8	56.52%	60.87%	47.83%	41.30%	54.35%
	label_9	32.50%	32.50%	22.50%	17.50%	32.50%

Table 8: Comparison of different unlearning methods on the Image Recognition(65) task across three clients.

From Table 8, our method exhibits a consistent pattern in the more complex Image Recognition(65) task. While the Zeroing route causes severe degradation on non-forgotten data (e.g., a 33.33% drop in class 9 on client1), indicating that our strategy effectively mitigates side effects on other data classes while preserving the unlearning effect.

Conclusion

We rethink the foundations of FU and show that two implicit assumptions, unfair global retraining and synthetic data partitions, have systematically inflated the reported effectiveness of FU mainstream technical routes. Building on this insight, we introduce FedCCCU, an evaluation framework that mirrors practical deployment conditions, and demonstrate through extensive experiments that mainstream tech-

nical routes remain fragile in both fairness and forgetting quality.

Although our study charts an initial path toward fair and deployable FU, key challenges remain in balancing unlearning precision and minimizing cross-client side effects. We encourage future work to enhance fairness-aware unlearning toward robust real-world deployment.

References

- Acar, D. A. E.; Zhao, Y.; Matas, R.; Mattina, M.; Whatmough, P.; and Saligrama, V. 2021. Federated Learning Based on Dynamic Regularization. In *International Conference on Learning Representations*. PMLR.
- Bourtole, L.; Chandrasekaran, V.; Choquette-Choo, C. A.; Jia, H.; Travers, A.; Zhang, B.; Lie, D.; and Papernot, N. 2021. Machine unlearning. In *2021 IEEE symposium on security and privacy (SP)*, 141–159. IEEE.
- Caldas, S.; Duddu, S. M. K.; Wu, P.; Li, T.; Konečný, J.; McMahan, H. B.; Smith, V.; and Talwalkar, A. 2018. LEAF: A Benchmark for Federated Settings. Preprint on arXiv, arXiv:1812.01097.
- Cao, X.; Jia, J.; Zhang, Z.; and Gong, N. Z. 2023. Fedrecover: Recovering from poisoning attacks in federated learning using historical information. In *2023 IEEE Symposium on Security and Privacy (SP)*, 1366–1383. IEEE.
- Fraboni, Y.; Van Waerebeke, M.; Scaman, K.; Vidal, R.; Kamani, L.; and Lorenzi, M. 2024. Sifu: Sequential informed federated unlearning for efficient and provable client unlearning in federated optimization. In *International Conference on Artificial Intelligence and Statistics*, 3457–3465. PMLR.
- Gao, X.; Ma, X.; Wang, J.; Sun, Y.; Li, B.; Ji, S.; Cheng, P.; and Chen, J. 2024. VeriFi: Towards Verifiable Federated Unlearning. *IEEE Transactions on Dependable and Secure Computing*, 21(6): 5720–5736.
- Goldman, E. 2020. An Introduction to the California Consumer Privacy Act (CCPA). SantaClaraUniv. LegalStudiesResearchPaper. Accessed: 2025-07-02.
- Gu, H.; Ong, W. K.; Chan, C. S.; and Fan, L. 2024. Ferrari: Federated Feature Unlearning via Optimizing Feature Sensitivity. *Advances in Neural Information Processing Systems*, 37: 24150–24180.
- Halimi, A.; Kadhe, S.; Rawat, A.; and Baracaldo, N. 2022. Federated Unlearning: How to Efficiently Erase a Client in FL? Preprint on arXiv, arXiv:2207.05521.
- He, Y.; Meng, G.; Chen, K.; He, J.; and Hu, X. 2021. DeepObliviate: A Powerful Charm for Erasing Data Residual Memory in Deep Neural Networks. Preprint on arXiv, arXiv:2105.06209.
- Hull, J. J. 1994. A Database for Handwritten Text Recognition Research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5): 550–554.
- Jang, J.; Yoon, D.; Yang, S.; Cha, S.; Lee, M.; Logeswaran, L.; and Seo, M. 2022. Knowledge Unlearning for Mitigating Privacy Risks in Language Models. Preprint on arXiv, arXiv:2210.01504.
- Jin, R.; Chen, M.; Zhang, Q.; and Li, X. 2023. Forgettable Federated Linear Learning With Certified Data Removal. Preprint on arXiv, arXiv:2306.02216.
- Karimireddy, S. P.; He, L.; and Jaggi, M. 2021. Learning from history for byzantine robust optimization. In *International conference on machine learning*, 5311–5319. PMLR.
- Krizhevsky, A.; and Hinton, G. 2009. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*.
- LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.
- Li, G.; Shen, L.; Sun, Y.; Hu, Y.; Hu, H.; and Tao, D. 2023. Subspace Based Federated Unlearning. Preprint on arXiv, arXiv:2302.12448.
- Li, N.; Zhou, C.; Gao, Y.; Chen, H.; Zhang, Z.; Kuang, B.; and Fu, A. 2025. Machine unlearning: Taxonomy, metrics, applications, challenges, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*.
- Li, X.; Jiang, M.; Zhang, X.; Kamp, M.; and Dou, Q. 2021. FedBN: Federated Learning on Non-IID Features via Local Batch Normalization. Preprint on arXiv, arXiv:2102.07623.
- Lin, Y.; Gao, Z.; Du, H.; Niyato, D.; Gui, G.; Cui, S.; and Ren, J. 2024. Scalable Federated Unlearning via Isolated and Coded Sharding. Preprint on arXiv, arXiv:2401.15957.
- Liu, G.; Ma, X.; Yang, Y.; Wang, C.; and Liu, J. 2021a. Federaser: Enabling efficient client-level data removal from federated learning models. In *2021 IEEE/ACM 29th International Symposium on Quality of Service (IWQOS)*, 1–10. IEEE.
- Liu, Y.; Ma, Z.; Yang, Y.; Liu, X.; Ma, J.; and Ren, K. 2021b. Revfrf: Enabling cross-domain random forest training with revocable federated learning. *IEEE Transactions on Dependable and Secure Computing*, 19(6): 3671–3685.
- Liu, Y.; Xu, L.; Yuan, X.; Wang, C.; and Li, B. 2022. The right to be forgotten in federated learning: An efficient realization with rapid retraining. In *IEEE INFOCOM 2022-IEEE conference on computer communications*, 1749–1758. IEEE.
- Liu, Z.; Jiang, Y.; Shen, J.; Peng, M.; Lam, K.-Y.; Yuan, X.; and Liu, X. 2024. A survey on federated unlearning: Challenges, methods, and future directions. *ACM Computing Surveys*, 57(1): 1–38.
- McCallum, A. K.; Nigam, K.; Rennie, J.; and Seymore, K. 2000. Automating the Construction of Internet Portals With Machine Learning. *Information Retrieval*, 3(2–3): 127–163.
- Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; Ng, A. Y.; et al. 2011. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, 4. Granada.
- Regulation, Protection. 2018. General Data Protection Regulation. *Intouch*, 25: 1–5.
- Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.;

et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115: 211–252.

Su, N.; and Li, B. 2023. Asynchronous federated unlearning. In *IEEE INFOCOM 2023-IEEE conference on computer communications*, 1–10. IEEE.

Wang, J.; Guo, S.; Xie, X.; and Qi, H. 2022. Federated unlearning via class-discriminative pruning. In *Proceedings of the ACM web conference 2022*, 622–632. ACM.

Wang, P.; Yan, Z.; Obaidat, M. S.; Yuan, Z.; Yang, L.; Zhang, J.; Wei, Z.; and Zhang, Q. 2023. Edge caching with federated unlearning for low-latency v2x communications. *IEEE Communications Magazine*, 62(10): 118–124.

Wu, C.; Zhu, S.; and Mitra, P. 2022. Federated Unlearning With Knowledge Distillation. Preprint on arXiv, arXiv:2201.09441.

Wu, X.; Li, J.; Xu, M.; Dong, W.; Wu, S.; Bian, C.; and Xiong, D. 2023. DepN: Detecting and Editing Privacy Neurons in Pretrained Language Models. Preprint on arXiv, arXiv:2310.20138.

Zhao, Y.; Wang, P.; Qi, H.; Huang, J.; Wei, Z.; and Zhang, Q. 2023. Federated Unlearning With Momentum Degradation. *IEEE Internet of Things Journal*, 11(5): 8860–8870.

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