RWFUBENCH: A CROSS-DOMAIN BENCHMARK FOR REVEALING COLLATERAL DAMAGE IN FEDERATED UNLEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Machine unlearning is essential for enforcing data deletion rights and is increasingly demanded in Federated Learning (FL). However, existing studies largely overlook a critical requirement, *Unlearning Independence*, which requires that uninvolved clients incur no extra unlearning computation or utility loss. This issue is further exacerbated in *Cross-Domain Non-IID* (CD-NonIID) settings, where clients share the same task and label space but draw data from different sources. Current evaluations, however, predominantly rely on simplified Pseudo-NonIID assumptions. To address this gap, we present a systematic benchmark named `RWFUBench` for the task of class unlearning in FL under CD-NonIID conditions. Our results reveal a tension between requester target suppression and retaining client utility. Retraining based methods retain high target accuracy consistent with cross-domain knowledge replenishment, while several model editing methods also reduce target accuracy on uninvolved clients. These findings show that aggregate evaluations can overlook collateral damage in federated unlearning. We further provide `FedCCCU` as a `Proof-of-Concept` mitigation strategy that reduces collateral damage, supporting future research on Unlearning Independence.

1 INTRODUCTION

With the use of machine learning, user data deletion has become an important requirement for privacy compliance. This demand is reinforced by legal frameworks like GDPR (European Parliament and Council of the European Union, 2016) and CCPA (Goldman, 2018), which grant users the right to data deletion. In response, Machine Unlearning (MU) (Bourtoule et al., 2021; Li et al., 2025) has been studied to realize the “right to be forgotten," and many methods remove the influence of selected data through retraining or targeted model changes. This setting is different from Federated Learning (FL). In FL, a global model is built by aggregating local updates from many clients, and the data of each client usually remain local. An unlearning request from one client can therefore affect the shared global model and the utility of other clients. Directly applying centralized MU to FL is also hard because the server cannot access all local data or repeat the whole training process for every deletion request. Federated Unlearning (FU) (Liu et al., 2024) is therefore needed to support data deletion in distributed systems while keeping the basic training process of FL.

Existing FU research mainly follows two lines based on overhead. Resource intensive methods cache historical gradients or intermediate model states, and then use these records to reconstruct or edit the model, as in FedEraser (Liu et al., 2021), VeriFi (Gao et al., 2024), SFU (Li et al., 2023), Amnesiac (Graves et al., 2021), Amnesiac-R (relabel) (Graves et al., 2021), KNOT (Su & Li, 2023), and FedRecovery (Zhang et al., 2023). These methods can improve forgetting, but they require large storage or extra computation that is hard to support in realistic FL systems. Some also need extra recovery or retraining steps after the request, which may involve other clients. Lightweight methods avoid historical caching and edit the current model through knowledge distillation, contrastive learning, pruning, or parameter manipulation, as in MoDe (Zhao et al., 2024), GA (Jang et al., 2023; Halimi et al., 2022), FedFilter (Wang et al., 2024), FCU (Deng et al., 2024), 2F2L (Jin et al., 2026), FedCDP (Wang et al., 2022), FedAU (Gu et al., 2024b), and DEPN (Wu et al., 2023). These methods reduce cost. Their edits can still change parameters that carry useful information for other clients, and most evaluations are conducted in small settings or under synthetic Non-IID partitions.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

These studies often leave two practical issues unclear. Unlearning Independence (UI) requires that an unlearning request should not force uninvolved clients to perform extra unlearning computation beyond their normal FL updates, and should not reduce their useful knowledge beyond normal model changes. Existing methods can violate this requirement. Retraining based methods may ask uninvolved clients to join extra training, while lightweight methods may edit parameters that are also important to retaining clients. This issue becomes clearer in CD-NonIID settings, where clients may share labels but differ in feature distributions. For example, two clients may share the same digit classes while one observes clean handwritten digits and another observes noisy street-view digits. Most current evaluations use Pseudo-NonIID splits from one dataset, which mainly change label ratios and do not test this feature shift. Such evaluations can hide whether unlearning is limited to the requester or whether it also changes useful knowledge held by retaining clients.

Figure 1 shows this gap between Pseudo-NonIID and cross-domain FU settings. In Pseudo-NonIID, clients usually come from the same dataset and differ mainly in class proportions. In CD-NonIID, clients from different domains, such as schools or banks, may share the same label space but rely on different features. We study FU by jointly considering UI and CD-NonIID. We build `RWFUBench` to evaluate class unlearning under CD-NonIID scenarios, where the requester asks to remove a target class and retaining clients should not face extra unlearning cost or lose unrelated utility. Our results show a clear tension. Some methods keep high accuracy on the forget set, while other methods reduce the target accuracy by also harming retaining clients. We further provide Federated Cross-Client-Constrained Unlearning (`FedCCCU`) as a `Proof-of-Concept` (PoC) strategy. It tests whether cross-client constraints can reduce collateral impact on retaining clients while suppressing target predictions on the requester.

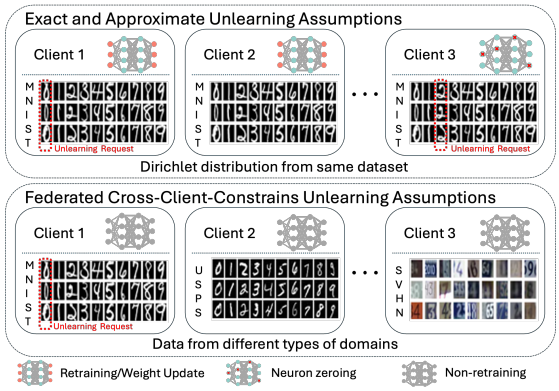

Figure 1: Federated unlearning in single-domain (top) and cross-domain (bottom) settings.

- We examine FU from the perspectives of Unlearning Independence and cross-domain heterogeneity, showing that existing FU methods can cause collateral damage and degrade useful knowledge of uninvolved clients, especially in CD-NonIID settings.
- We build a systematic `RWFUBench` of representative FU methods under CD-NonIID, revealing a tension between requester target suppression and retaining client utility.
- We propose `FedCCCU`, a PoC strategy that reduces retaining client knowledge loss while suppressing requester target predictions, demonstrating a direction toward improving Unlearning Independence in FU systems. Code is available at https://anonymous.4open.science/r/alksdjfo/

2 Related Work

Existing FU methods mainly follow two lines that are also used in our benchmark. Resource intensive methods, such as FedEraser (Liu et al., 2021), Amnesiac (Graves et al., 2021), and FedRecovery (Zhang et al., 2023), store past updates or model states and use them to remove requested data. Some require retaining clients to join new training or recovery steps, which conflicts with Process Independence. Lightweight methods, such as MoDe (Zhao et al., 2024), FedFilter (Wang et al., 2024), FedCDP (Wang et al., 2022), Ferrari (Gu et al., 2024a), FedAU (Gu et al., 2024b), FCU (Deng et al., 2024), and DEPN (Wu et al., 2023), edit the current model by distillation, gradient update, pruning, or parameter change. These methods reduce storage and training cost, but their edits can affect knowledge used by retaining clients.

Table 1 outlines assumptions in prior FU studies. Most methods are tested on IID or Pseudo-NonIID data, where clients are built from the same dataset. This setting does not test the feature shift present in CD-NonIID settings. Several methods also rely on multi client retraining, which is not aligned with UI. We therefore keep representative methods from both lines in the main benchmark and provide a

broader method summary in Appendix D. This comparison motivates `RWFUBench`, which evaluates FU in CD-NonIID settings and checks both forgetting and the utility of retaining clients.

3 Problem Formulation

3.1 The Unlearning Independence Dilemma

In a federated system, the global risk $F(\mathbf{w})$ is minimized across K clients and is written as

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

$\mathbf{w}$ denotes the model parameters, $\mathcal{D}_k$ is the data distribution of client k, and D_k is its local training dataset with n_k samples. The total sample count is $n = \sum_{k=1}^{K} n_k$. The loss is $\ell(\mathbf{w}, \mathbf{x}, y)$, where $\mathbf{x}$ is the input and y is the label. Training continues until the validation error falls below ε. The round when this condition is met is T_0, yielding $\mathbf{w}_{T_0}$.

Table 1: FU method assumptions. MCR means retraining with other clients. Pseudo-NonIID uses one dataset, while CD-NonIID uses distinct datasets.

Method	Year	MCR	Data
FedEraser	2021	✔	IID / Pseudo-NonIID
Amnesiac	2021	✔	IID / Pseudo-NonIID
DeepObliviate	2021	✔	Pseudo-NonIID
RevFRF	2021	✔	Pseudo-NonIID
Wu et al	2022	✔	Pseudo-NonIID
Liu et al	2022	✔	Pseudo-NonIID
FedCDP	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
FedAU	2024	✘	IID / Pseudo-NonIID
FCU	2024	✘	Pseudo-NonIID
SIFU	2024	✔	Pseudo-NonIID
FUSED	2025	✔	Pseudo-NonIID
FedOSD	2025	✔	IID / Pseudo-NonIID
`FedCCCU` (Ours)	-	✘	**CD-NonIID**

When a subset of clients C_{req} requests the removal of data D_u, a common reference is to generate $\mathbf{w}_{\text{gold}}$ by retraining on $D_{\text{ret}} = (\bigcup_{k=1}^{K} D_k) \setminus D_u$. Let C_{ret} denote clients that do not request unlearning. Process Independence requires that the request should not add unlearning triggered computation for any client $k \in C_{\text{ret}}$. This does not prevent normal FL updates, since clients may still join regular model training. It only excludes extra work caused by the deletion request. We write this condition as $\Delta\text{Cost}_k = \text{Cost}_k^{\text{actual}} - \text{Cost}_k^{\text{normal}} = 0$, where $\text{Cost}_k^{\text{actual}}$ is the cost after the request and $\text{Cost}_k^{\text{normal}}$ is the cost under normal FL updates. Methods that require extra recovery or retraining from retaining clients have $\Delta\text{Cost}_k > 0$ and violate Process Independence.

To avoid this cost, lightweight methods often derive an edited model $\mathbf{w}_{\text{edit}}$ from $\mathbf{w}_{T_0}$. This may violate Utility Independence. Let $A_k(\mathbf{w})$ denote accuracy on client k's local test set D_k^{test}, and $A_{k,c}(\mathbf{w})$ accuracy on its class c subset. Utility Independence requires preserving retaining client utility at both levels. An observed overall loss is $A_k(\mathbf{w}_{\text{edit}}) < A_k(\mathbf{w}_{T_0})$, with the analogous comparison for $A_{k,c}$. These comparisons measure changes from the initial model, including any intervening normal training variation.

Therefore, FU evaluation should consider unlearning efficacy alongside the two UI requirements. Process Independence concerns extra unlearning computation, while Utility Independence concerns retaining client utility. For class unlearning, we measure requester target suppression through the reduction in $A_{1,c}$ and assess it alongside privacy leakage and retaining client utility. Target accuracy alone does not establish whether the influence of deleted training data has been removed. Requester only retraining in Section 4 can preserve Process Independence while leaving high target accuracy supported by other clients.

3.2 Pseudo-NonIID and Cross-Domain Non-IID

Existing FU research largely relies on a synthetic Pseudo-NonIID assumption, which usually partitions one dataset with a Dirichlet distribution. This setting changes the label distribution $P_k(y)$ across clients, so it can create $P_k(y) \neq P_j(y)$ for clients k and j. Yet samples of the same class still come from the same source, which often keeps the conditional feature distributions similar, $P_k(x|y) \approx P_j(x|y)$. This does not capture many real-world settings in which clients collect data from different institutions, devices, or environments. For instance, two hospitals may diagnose the same disease

y, while hospital k uses CT images and hospital j uses noisier X-ray images. The label meaning is the same, but the feature distributions $P_k(x|y)$ and $P_j(x|y)$ can differ. We refer to this setting as Cross-Domain Non-IID (CD-NonIID), characterized by

$$P_k(x|y) \neq P_j(x|y) \quad \text{for } k \neq j \tag{2}$$

CD-NonIID can make UI harder to satisfy because target knowledge and domain specific features may use related model parameters, which increases the risk of collateral damage to uninvolved clients.

4 RWFUBENCH

We propose `RWFUBench` to evaluate FU under CD-NonIID. Unlike prior settings that partition a single dataset, it treats each client as an autonomous data silo with heterogeneous features and labels. Each evaluated method applies its retraining or editing procedure to address the request. In Retrain-PI, only the requester retrains from scratch, while retaining clients continue normal training from their existing weights. The server aggregates updates until convergence. Retaining prior weights does not freeze retaining client training; only the requester retrains in the Retrain-PI baseline.

4.1 DATASET OVERVIEW AND SELECTION RATIONALE

We construct `RWFUBench` to keep task consistency while creating clear feature divergence. DomainNet (Peng et al., 2019) and PACS (Li et al., 2017) are standard transfer learning datasets, and some FL studies adapt them. We do not use them as main tasks because their complexity can make the initial global model underfit or fluctuate. This instability makes it hard to tell whether an accuracy drop after unlearning comes from the unlearning method or from the base model. Instead, we use MNIST10 (LeCun et al., 1998), SVHN (Netzer et al., 2011), USPS (Hull, 1994), CIFAR10/100 (Krizhevsky, 2009), and ImageNet (Russakovsky et al., 2015). These datasets support stable initial convergence and still create clear domain gaps. We keep only overlapping labels to preserve task consistency across domains, yielding 9 shared labels for CIFAR10 and ImageNet and 65 shared labels for CIFAR100 and ImageNet. IR-65 is harder because it has more shared classes and lower initial accuracy, so it tests FU methods under a stronger controlled feature shift. More details are in Appendix E.

These public datasets provide a reproducible CD-NonIID setting: MNIST, SVHN, and USPS differ in background, clarity, and visual style, while CIFAR and ImageNet differ in resolution, background, and visual detail. The clients share labels but differ in feature distributions.

4.2 DATA DISTRIBUTION STRATEGY

To represent cross-domain feature heterogeneity, we design a CD-NonIID strategy (allocations in Table 2) that maintains task consistency while inducing feature heterogeneity.

Table 2: CD-NonIID experimental configuration. HDR: Handwritten Digit Recognition; IR-X: Image Recognition with X shared classes (e.g., IR-9 and IR-65).

Scenario	Task	Client	Dataset	Res.	Model
CD-NonIID	HDR	C_1-C_3	MNIST10	28×28	CNN
		C_4-C_6	SVHN	32×32	
		C_7-C_9	USPS	16×16	
	IR-9	C_1-C_5	CIFAR10	32×32	ResNet18
		C_6-C_{10}	ImageNet	224×224	
	IR-65	C_1-C_5	CIFAR100	32×32	ResNet32
		C_6-C_{10}	ImageNet	224×224	

For HDR, clients are grouped by distinct source domains (MNIST, SVHN, USPS) with intra-group partitions via a Dirichlet distribution. For IR tasks (IR-9/IR-65), we split clients between low-resolution CIFAR and high-resolution ImageNet domains. To process these varying inputs without artifact-inducing resizing, we incorporate an AdaptiveAvgPool2d layer to strictly align feature dimensions. This "task-aligned but domain-divergent" setup introduces multi-dimensional shifts (e.g., resolution, texture) typical of real-world cross-organization systems. In all scenarios, the unlearning request is uniquely initiated by Client 1 (C_1) targeting 'label_0' (e.g., 'airplane').

Figure 2 schematically contrasts the data assumptions. The axes show clients (x), distribution regimes (y), and class counts (z). The chart illustrates class coverage across regimes; the distribution axis is categorical and does not quantify feature heterogeneity. In CD-NonIID, clients can share class coverage while differing in source domain, as specified in Table 2.

Table 3: Global test accuracy (Acc_{global}) and requester target class test accuracy ($Acc_{c_f,f} = A_{1,c}$) before and after unlearning under CD-NonIID. **Red shading** marks high post-unlearning target accuracy for marked baselines.

Method	HDR Task				IR-9 Task				IR-65 Task			
	Acc_{global}		$Acc_{c_f,f}$		Acc_{global}		$Acc_{c_f,f}$		Acc_{global}		$Acc_{c_f,f}$	
	Pre	Post	Pre	Post	Pre	Post	Pre	Post	Pre	Post	Pre	Post
Retrain-PI	96.82%	95.3%	99.65%	99.82%	90.77%	88.46%	94.33%	82.77%	72.99%	69.34%	62.86%	45.71%
Amnesiac-R	95.84%	95.3%	99.65%	98.42%	90.77%	88.62%	94.33%	83.67%	69.9%	69.56%	62.86%	51.43%
Amnesiac	95.63%	55.82%	99.82%	10.68%	79.84%	20.55%	87.00%	0.00%	56.55%	4.08%	75.00%	0.00%
Ferrari	94.48%	86.13%	99.65%	52.67%	79.12%	65.71%	83.39%	54.04%	58.90%	7.55%	70.83%	0.00%
FedCDP	95.68%	80.37%	99.82%	59.19%	79.75%	60.12%	85.00%	35.50%	58.11%	43.67%	75.00%	35.00%
FedAU	95.24%	87.51%	99.65%	0.00%	81.35%	73.07%	82.31%	0.00%	57.64%	56.64%	68.57%	0.00%
FCU	90.22%	59.76%	96.85%	0.00%	85.27%	11.14%	85.26%	0.91%	63.46%	39.97%	75.00%	0.00%
DEPN	95.3%	88.04%	99.65%	1.93%	90.77%	73.66%	94.33%	7.94%	69.46%	60.31%	62.86%	0.00%

4.3 Benchmark Experiments

Prior studies report FU results in IID or Pseudo-NonIID settings, but their behavior in CD-NonIID environments is less clear. In these settings, we assess UI by checking whether one client's unlearning request reduces the utility of retaining clients. This check is needed because global accuracy can hide client utility loss. Table 3 presents benchmark results under CD-NonIID conditions, evaluating representative FU approaches (Retrain-PI, Amnesiac, Amnesiac-R (relabel), Ferrari, FedCDP, FedAU, FCU, and DEPN). To satisfy Process Independence for Retrain-PI, only the requesting client retrains from scratch. Uninvolved clients resume normal training from previous weights, so no extra unlearning computation is added to them. This setting separates normal FL updates from extra deletion request overhead.

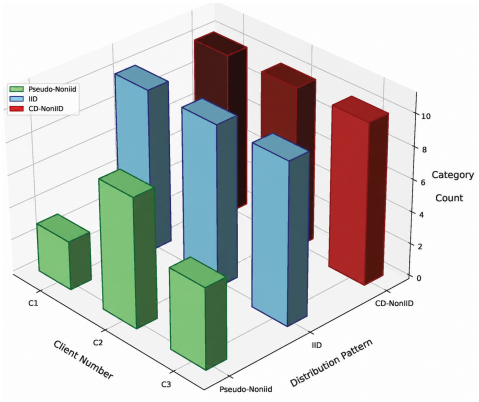

Figure 2: Schematic illustration of client class coverage under three distribution regimes. The categorical distribution axis does not measure feature shift.

We adapt existing methods to follow Process Independence. This prevents the system from asking uninvolved clients to spend extra computation on unrelated unlearning requests. We modify Amnesiac to operate locally. The server revokes historical parameter updates only for the requester, and no compensatory fine tuning is taken from uninvolved clients. FedCDP needs similar limits. We restrict the TF-IDF pruning scores and later fine tuning to the local requester. These changes align the evaluated procedures with the PI requirement in Section 3.1. Pre and Post evaluate each method's own model before and after its unlearning procedure. The method specific Pre values differ, so comparisons consider changes from each starting point alongside Post accuracy.

The red shading in Table 3 shows that Retrain-PI and Amnesiac-R keep high accuracy on the requester's forget set. Under cross-domain label sharing, this output level accuracy cannot by itself tell whether the requester's influence is removed or whether retaining clients still supply related target knowledge during aggregation. We refer to this possible effect as Cross-Domain Knowledge Replenishment. The observed pattern is consistent with this explanation but does not establish causality. It motivates the need to inspect both requester forgetting and retaining client utility.

Not all lightweight methods cause a large drop in global accuracy. FedAU and DEPN appear to provide a better tradeoff in Table 3. For instance, on IR-65, FedAU reduces forget set accuracy to 0.00% while the global accuracy changes from 57.64% to 56.64%. Yet global accuracy and requester forget set accuracy are not enough to assess UI. They do not show whether the same target knowledge is also removed from uninvolved clients that hold related data. A method can look stable at the global level while still harming a subset of clients or labels. We report global retraining results in Appendix F as a reference, while noting that it requires extra participation from uninvolved clients.

Figure 3 shows this hidden cost by comparing client level target accuracy. For FedAU and DEPN, the orange bar for the requester C1 drops to zero or a low level, showing strong target suppression. The

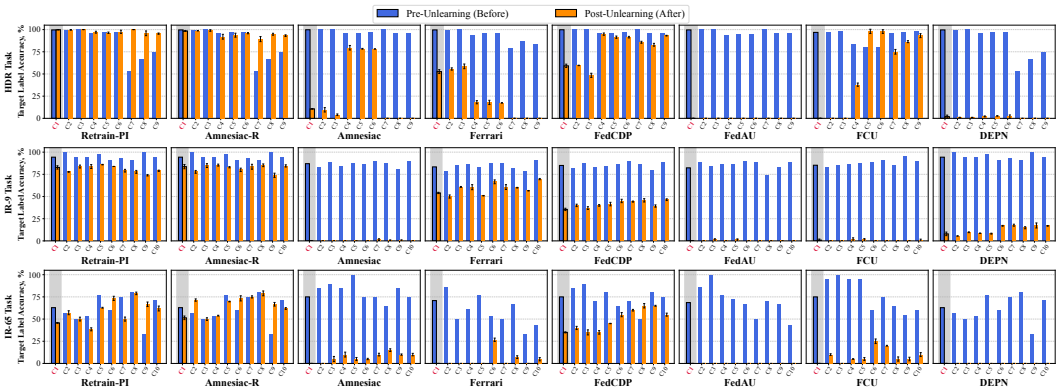

Figure 3: Target class accuracy across clients and eight methods before and after unlearning. Post-unlearning bars show the mean over three seeds. Error bars show the sample standard deviation. Each row shows HDR with 9 clients, IR-9 with 10 clients, and IR-65 with 10 clients.

Table 4: MIA Attack Success Rate (ASR) on the unlearned target class. Values near 50% indicate lower leakage under this attack.

Task	Attack Success Rate (ASR) (%)							
	Retrain-PI	**Amnesiac-R**	**Amnesiac**	**Ferrari**	**FedCDP**	**FedAU**	**FCU**	**DEPN**
HDR	78.33	76.84	50.89	61.32	61.84	51.66	52.54	51.49
IR-9	72.25	74.72	52.36	59.33	56.13	50.85	52.56	52.24
IR-65	67.49	65.32	50.89	50.32	52.91	52.87	53.02	51.67

orange bars for uninvolved clients such as C2 and C3 also drop close to zero in several cases. These retaining client losses conflict with the Utility Independence requirement in Section 3.1. The key issue is where the target loss occurs across clients, not only whether it occurs.

This pattern also appears in other lightweight methods. In Amnesiac and Ferrari, the orange bars for many uninvolved clients are shorter than their blue bars, indicating target accuracy loss on retaining clients. This helps explain the global accuracy drops in Table 3. In contrast, Retrain-PI and Amnesiac-R retain higher target accuracy on C1 and retaining clients, indicating weaker requester suppression. Across all 232 client results from the eight methods, the mean sample standard deviation is 0.94% and the maximum is 2.91%. The complete client-wise values are reported in Appendix Table 13. The figure shows a tension between requester target suppression and retaining client utility. Appendix G extends this evaluation to client level unlearning.

To further assess forgetting and privacy leakage, we conduct Membership Inference Attacks (MIA) (Shokri et al., 2017) on the unlearned target class. Table 4 shows that Retrain-PI and Amnesiac-R have Attack Success Rates above 70% on HDR and IR-9, and above 65% on IR-65. These results indicate residual leakage under this attack. Lightweight methods such as FedAU produce ASR values near random guessing, but Figure 3 shows that they may also reduce retaining target accuracy. The MIA result is attack specific evidence, not a formal privacy claim. We combine it with forget set accuracy and client level target accuracy to assess the tradeoff between leakage and retaining utility.

5 Discussion on Potential Solution

The benchmark analysis in Section 4 shows a gap in current FU research. Existing methods can leave high forget set accuracy or reduce retaining client utility. This section studies a PoC mitigation path to test whether retaining client information can reduce this tension. To explore this possibility, we introduce `FedCCCU`. Figure 4 shows that `FedCCCU` edits selected neurons under cross-client constraints so that the unlearning effect is more local to the requester. The method assumes that compact sensitivity summaries have been computed and collected before the deletion request. Preparation requires attribution computation and communication, while subsequent edits are made at the server without extra retaining client work. Its PI claim therefore applies to the request stage with

these summaries already available. This design tests whether cross-client constraints can reduce side effects in `RWFUBench`.

5.1 Identification of Key Neurons

To implement this PoC, we need a simple way to locate neurons related to the target class. For any neuron u_l^q at layer l with index q, we estimate its contribution to classifying a sample x_i as the target class c. We use q for the neuron index to avoid confusion with the client index k used earlier. The model parameter is kept implicit in $P(c|x_i)$ for readability. We compute the cumulative gradient of the class probability as the activation changes from 0 to its original value β_l^q. This contribution is called the Attribution Score and is written as

$$\mathrm{Att}(u_l^q, x_i) = \int_0^{\beta_l^q} \frac{\partial P(c|x_i, \alpha_l^q)}{\partial \alpha_l^q} d\alpha_l^q \tag{3}$$

Here β_l^q is the original activation value of neuron u_l^q for input x_i. The term $P(c|x_i, \alpha_l^q)$ is the probability of class c when the activation is set to α_l^q. The derivative $\frac{\partial P(\cdot)}{\partial \alpha_l^q}$ measures how the class probability changes with this activation. Directly computing the integral is costly, so we approximate it with a Riemann sum using m steps as follows

$$\mathrm{Att}(u_l^q, x_i) \approx \frac{\beta_l^q}{m} \sum_{j=1}^{m} \frac{\partial P(c|x_i, \alpha_l^q)}{\partial \alpha_l^q}\Big|_{\alpha_l^q = \frac{j}{m}\beta_l^q} \tag{4}$$

where $\frac{j}{m}\beta_l^q$ is the activation value at step j.

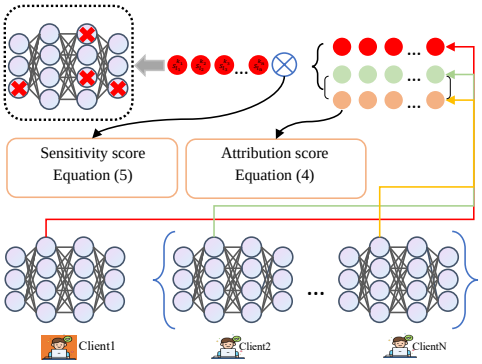

Figure 4: Overview of `FedCCCU`. Clients send compact neuron sensitivity scores. The server compares requester and retaining client scores, then edits requester-dominant neurons to reduce collateral impact.

The client then calculates these attribution scores for all neurons across samples in its local dataset D_{client}. To reduce sample noise, we average these scores and obtain a sensitivity score S_l^q for each neuron u_l^q as follows

$$S_l^q = \frac{1}{|D_{client}|} \sum_{x_i \in D_{client}} \mathrm{Att}(u_l^q, x_i) \tag{5}$$

where $|D_{client}|$ is the number of samples in the local dataset. For class c, this score measures how strongly the neuron is related to that class on this client. During preparation, each client identifies the indices (l, q) and scores S_l^q of its Top-N sensitive neurons for each class. It sends these summaries to the server for later use in the constraint step. The raw data remain local. The server only receives which neurons are sensitive and how large their scores are. It does not need samples, gradients from all training rounds, or extra local fine tuning from retaining clients. The compact lists limit communication volume, but their size alone does not establish Process Independence. Appendix C reports the size of these summaries.

5.2 Cross-Client Constraint via Dominant Neurons

To use UI in the PoC, we define a metric that compares the requester's need to edit a neuron with the retaining clients' need to keep it. We call this metric the Utility Conflict Ratio (R). For a target class c, we use the sensitivity scores above to compare the relative importance of each neuron u_l^q. The metric is simple by design. It is meant to expose whether a target related neuron is mostly used by the requester or is also important to retaining clients.

Let $\mathcal{S}_{\text{req}}(u_l^q)$ denote the sensitivity of neuron u_l^q for the forgetting client C_{req}. We compare it with the retaining clients' reported sensitivities and define $\mathcal{S}_{\max}(u_l^q) = \max_{j \in C_{\text{ret}}} \mathcal{S}_j(u_l^q)$. A score absent from a client's Top-N list is represented by zero. Thus, $\mathcal{S}_{\max}$ is a sparse estimate and can understate

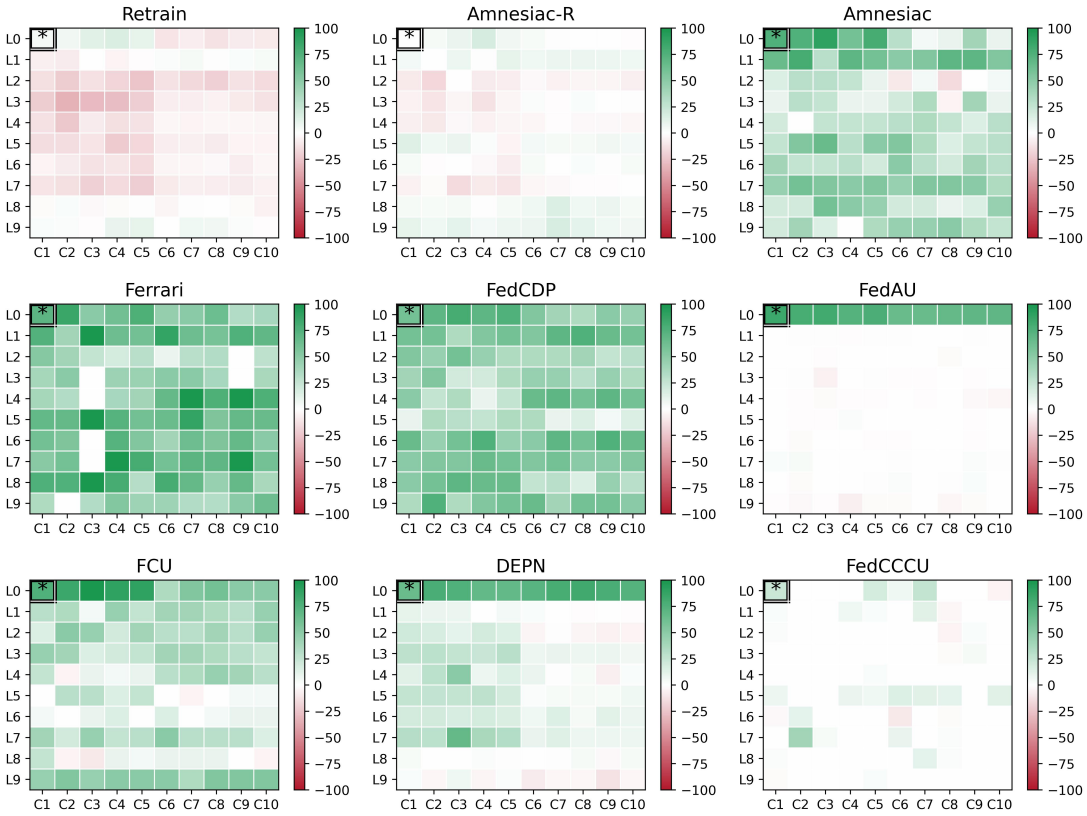

Figure 5: Client-wise accuracy shifts on IR-65 ($\Delta = \text{Acc}_{\text{before}} - \text{Acc}_{\text{after}}$) in %. We show L0–L9 because space is limited for the 65 classes. **Green** ($\Delta > 0$) indicates accuracy loss, **red** ($\Delta < 0$) gain, and **white** ($\Delta \approx 0$) stability. Green at the target (C1, L0) indicates target suppression; green on retaining clients indicates collateral loss. Gains do not constitute utility loss.

retaining client dependence when relevant scores are omitted. The numerator estimates retaining client dependence, while the denominator measures requester dependence. The conflict ratio R is defined as

$$R(u_l^q) = \frac{\text{ReLU}(\mathcal{S}_{\text{max}}(u_l^q))}{\text{ReLU}(\mathcal{S}_{\text{req}}(u_l^q)) + \epsilon} \tag{6}$$

Here ReLU keeps only positive sensitivity scores, and ϵ is a small constant used for numerical stability. A high Utility Conflict Ratio indicates greater estimated retaining sensitivity relative to requester sensitivity, so editing the neuron may harm retaining utility. Lower ratios prioritize neurons with lower estimated relative conflict. The term “Dominant Neuron" describes this ranking preference and does not imply exclusive use by the requester. Thus, R is a practical pre-editing ranking score.

We then use a Rank-Based Selection strategy. The requester's N candidate neurons are sorted by their Utility Conflict Ratio in ascending order. The server selects the P neurons with the lowest ratios, where $P \leq N$, and zeroes out the weights at their indices (l, q). Selection does not impose $R < 1$ and therefore does not guarantee greater requester sensitivity for every edited neuron. This targeted pruning aims to reduce target predictions while limiting estimated conflict with retaining clients. More complex edits can be built on the same ranking in future work.

5.3 PoC Validation and Analysis

We evaluate `FedCCCU` on IR-9 and IR-65 under CD-NonIID. By prioritizing neurons with lower estimated Utility Conflict Ratios, it directs edits toward requester target predictions while limit-

ing changes to neurons used by retaining clients. The results support this strategy: Figure 10 in Appendix K reports the aggregate comparison, while Figure 5 shows client- and label-level effects.

Reducing accuracy at C1 L0 while keeping other cells unchanged is an ideal for localized editing. Utility Independence concerns losses on retaining clients, while the target drop measures requester suppression. Retrain-PI leaves the target cell relatively unchanged, which is consistent with the knowledge replenishment explanation. FedAU suppresses the target more strongly, but also reduces target accuracy for retaining clients such as C2 and C3. In contrast, `FedCCCU` produces a target drop with smaller side effects in this comparison, consistent with its preference for lower estimated conflict. Average accuracy alone can miss these client level effects. The result provides PoC evidence that cross-client constraints can reduce collateral loss.

Appendix H further shows that the larger accuracy change on C1 than on retaining clients persists across six target classes.

Parameter sensitivity. Table 5 varies N and P on IR-65 while keeping the other settings unchanged.

Table 5: Parameter sensitivity of `FedCCCU` on IR-65. Accuracy values are in %. The N sweep fixes $P = 2$, and the P sweep fixes $N = 20$. Ret. L0 is the mean target class accuracy of C2–C10. Bold values show the main setting.

			Candidate neurons N ($P = 2$)					Pruned neurons P ($N = 20$)					
Task	**Metric**	**Pre**	**20**	**30**	**50**	**70**	**100**	**1**	**2**	**4**	**8**	**16**	**20**
IR-65	Global	69.46	**68.90**	68.82	68.79	68.63	67.83	69.28	**68.90**	68.80	67.51	66.07	56.08
	C1-L0	62.86	**45.16**	43.72	39.51	38.83	38.06	45.37	**45.16**	41.96	37.42	36.94	31.43
	Ret. L0	71.11	**68.33**	68.17	65.31	65.04	63.89	70.36	**68.33**	61.67	58.89	45.08	43.61

The two parameters control different aspects of the edit. With P fixed, N changes the candidate pool used by the Utility Conflict Ratio without increasing the number of edited neurons. With N fixed, larger P strengthens requester suppression alongside greater retaining client loss. The main setting ($N = 20$, $P = 2$) reduces requester target accuracy by 17.70% and retaining target accuracy by 2.78%, illustrating the tradeoff between suppression and preservation. Table 15 in Appendix L reports the corresponding IR-9 analysis.

Model architectures. With only architecture changed, C1 target accuracy drops exceed the C2–C10 average, while retaining client accuracy remains above C1-L0 after unlearning on both tasks. This pattern shows that the sensitivity comparison and Utility Conflict Ratio ranking apply to both residual and VGG11-BN models.

Table 6: Results across model architectures. Values show accuracy before and after unlearning in %.

Task	**Model**	**Global**	**C1-L0**	**Ret. L0**
IR-9	ResNet18	90.77 / 81.82	94.33 / 18.00	93.23 / 71.04
	VGG11-BN	86.41 / 73.41	94.61 / 1.73	90.42 / 51.68
IR-65	ResNet32	69.46 / 68.90	62.86 / 45.16	71.11 / 68.33
	VGG11-BN	66.02 / 64.52	70.34 / 25.33	75.02 / 60.56

`FedCCCU` is faster than retraining based baselines and close in runtime to practical model editing methods (Appendix K, Table 14). Appendix C reports ASR values of approximately 50–53% for `FedCCCU`, near random guessing under the evaluated attack. This privacy result is attack specific and is used together with the accuracy results. Appendix A and B provide supporting feasibility and additional experimental comparisons.

6 Conclusion

`RWFUBench` evaluates class-level FU under realistic CD-NonIID while considering Process and Utility Independence together. The results show that retraining-based methods can retain high target accuracy, whereas several lightweight methods suppress the target at the cost of retaining-client utility, failures that Pseudo-NonIID evaluation can obscure. `FedCCCU` provides a PoC that limits collateral effects through cross-client constraints. These findings motivate CD-NonIID evaluation of both forgetting and Unlearning Independence; arbitrary sample-level requests, text and audio data, faster intervention, and finer unlearning targets remain for future work.

AI USE STATEMENT

In this work, we have not used generative AI tools for literature retrieval or discovery, research ideation or execution, producing initial drafts of manuscript sections, generating synthetic datasets, or proving mathematical claims, and the remaining required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools to improve the grammar, clarity, concision, and presentation of text written by us. We reviewed all AI-assisted work and checked each suggested edit against our intended technical meaning, equations, experimental results, citations, and claims. We take responsibility for the final content of this work, including any text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

D. A. E. Acar, Y. Zhao, R. Matas, M. Mattina, P. Whatmough, and V. Saligrama. Federated learning based on dynamic regularization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=B7v4QMR6Z9w.

L. Bourtoule, V. Chandrasekaran, C. A. Choquette-Choo, H. Jia, A. Travers, B. Zhang, D. Lie, and N. Papernot. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pp. 141–159, 2021. doi: 10.1109/SP40001.2021.00019.

S. Caldas, S. M. K. Duddu, P. Wu, T. Li, J. Konečný, H. B. McMahan, V. Smith, and A. Talwalkar. LEAF: A benchmark for federated settings, 2018. URL https://arxiv.org/abs/1812.01097. Preprint on arXiv.

Z. Deng, L. Luo, and H. Chen. Enable the right to be forgotten with federated client unlearning in medical imaging. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2024*, volume 15010 of *Lecture Notes in Computer Science*, pp. 240–250, 2024. doi: 10.1007/978-3-031-72378-0_23.

European Parliament and Council of the European Union. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 april 2016 (General Data Protection Regulation). Official Journal of the European Union, L 119, pp. 1–88, 2016. URL https://eur-lex.europa.eu/eli/reg/2016/679/oj.

Yann Fraboni, Martin Van Waerebeke, Kevin Scaman, Richard Vidal, Laetitia Kameni, and Marco Lorenzi. SIFU: Sequential informed federated unlearning for efficient and provable client unlearning in federated optimization. In *Proceedings of the 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pp. 3457–3465. PMLR, 2024. URL https://proceedings.mlr.press/v238/fraboni24a.html.

X. Gao, X. Ma, J. Wang, Y. Sun, B. Li, S. Ji, P. Cheng, and J. Chen. VeriFi: Towards verifiable federated unlearning. *IEEE Transactions on Dependable and Secure Computing*, 21(6):5720–5736, 2024. doi: 10.1109/TDSC.2024.3382321.

Eric Goldman. An introduction to the California Consumer Privacy Act (CCPA). *SSRN Electronic Journal*, 2018. doi: 10.2139/ssrn.3211013. URL https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3211013.

L. Graves, V. Nagisetty, and V. Ganesh. Amnesiac machine learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(13):11516–11524, 2021. doi: 10.1609/aaai.v35i13.17371.

H. Gu, W. K. Ong, C. S. Chan, and L. Fan. Ferrari: Federated feature unlearning via optimizing feature sensitivity. In *Advances in Neural Information Processing Systems*, volume 37, pp. 24150–24180, 2024a. doi: 10.52202/079017-0761.

H. Gu, G. Zhu, J. Zhang, X. Zhao, Y. Han, L. Fan, and Q. Yang. Unlearning during learning: An efficient federated machine unlearning method. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pp. 4035–4043, 2024b. doi: 10.24963/ijcai.2024/446.

A. Halimi, S. Kadhe, A. Rawat, and N. Baracaldo. Federated unlearning: How to efficiently erase a client in FL?, 2022. URL https://arxiv.org/abs/2207.05521. Preprint on arXiv.

Y. He, G. Meng, K. Chen, J. He, and X. Hu. DeepObliviate: A powerful charm for erasing data residual memory in deep neural networks, 2021. URL https://arxiv.org/abs/2105.06209. Preprint on arXiv.

J. J. Hull. A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):550–554, 1994. doi: 10.1109/34.291440.

J. Jang, D. Yoon, S. Yang, S. Cha, M. Lee, L. Logeswaran, and M. Seo. Knowledge unlearning for mitigating privacy risks in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14389–14408, 2023. doi: 10.18653/v1/2023.acl-long.805.

R. Jin, M. Chen, Q. Zhang, and X. Li. Forgettable federated linear learning with certified data unlearning. *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–10, 2026. doi: 10.1109/TNNLS.2026.3683398.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5543–5551, 2017. doi: 10.1109/ICCV.2017.591.

G. Li, L. Shen, Y. Sun, Y. Hu, H. Hu, and D. Tao. Subspace based federated unlearning, 2023. URL https://arxiv.org/abs/2302.12448. Preprint on arXiv.

N. Li, C. Zhou, Y. Gao, H. Chen, Z. Zhang, B. Kuang, and A. Fu. Machine unlearning: Taxonomy, metrics, applications, challenges, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*, 36(12):24700–24721, 2025. doi: 10.1109/TNNLS.2025.3530988.

X. Li, M. Jiang, X. Zhang, M. Kamp, and Q. Dou. FedBN: Federated learning on Non-IID features via local batch normalization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=6YEQUn0QICG.

Y. Lin, Z. Gao, H. Du, D. Niyato, G. Gui, S. Cui, and J. Ren. Scalable federated unlearning via isolated and coded sharding. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pp. 4551–4559, 2024. doi: 10.24963/ijcai.2024/503.

G. Liu, X. Ma, Y. Yang, C. Wang, and J. Liu. FedEraser: Enabling efficient client-level data removal from federated learning models. In *2021 IEEE/ACM 29th International Symposium on Quality of Service (IWQOS)*, pp. 1–10, 2021. doi: 10.1109/IWQOS52092.2021.9521274.

Y. Liu, Z. Ma, Y. Yang, X. Liu, J. Ma, and K. Ren. RevFRF: Enabling cross-domain random forest training with revocable federated learning. *IEEE Transactions on Dependable and Secure Computing*, 19(6):3671–3685, 2022a. doi: 10.1109/TDSC.2021.3104842.

Y. Liu, L. Xu, X. Yuan, C. Wang, and B. Li. The right to be forgotten in federated learning: An efficient realization with rapid retraining. In *IEEE INFOCOM 2022 – IEEE Conference on Computer Communications*, pp. 1749–1758, 2022b. doi: 10.1109/INFOCOM48880.2022.9796721.

Z. Liu, Y. Jiang, J. Shen, M. Peng, K. Lam, X. Yuan, and X. Liu. A survey on federated unlearning: Challenges, methods, and future directions. *ACM Computing Surveys*, 57(1):1–38, 2024. doi: 10.1145/3679014. Article 2.

Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011. URL https://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf.

Z. Pan, Z. Wang, C. Li, K. Zheng, B. Wang, X. Tang, and J. Zhao. Federated unlearning with gradient descent and conflict mitigation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39 (19):19804–19812, 2025. doi: 10.1609/aaai.v39i19.34207.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1406–1415, 2019. doi: 10.1109/ICCV.2019.00149.

O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and Fei-Fei Li. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 3–18, 2017. doi: 10.1109/SP.2017.41.

N. Su and B. Li. Asynchronous federated unlearning. In *IEEE INFOCOM 2023 – IEEE Conference on Computer Communications*, pp. 1–10, 2023. doi: 10.1109/INFOCOM53939.2023.10229075.

J. Wang, S. Guo, X. Xie, and H. Qi. Federated unlearning via class-discriminative pruning. In *Proceedings of the ACM Web Conference 2022*, pp. 622–632. ACM, 2022. doi: 10.1145/3485447.3512222.

P. Wang, Z. Yan, M. S. Obaidat, Z. Yuan, L. Yang, J. Zhang, Z. Wei, and Q. Zhang. Edge caching with federated unlearning for low-latency V2X communications. *IEEE Communications Magazine*, 62(10):118–124, 2024. doi: 10.1109/MCOM.001.2300272.

C. Wu, S. Zhu, and P. Mitra. Federated unlearning with knowledge distillation, 2022. URL `https://arxiv.org/abs/2201.09441`. Preprint on arXiv.

X. Wu, J. Li, M. Xu, W. Dong, S. Wu, C. Bian, and D. Xiong. DEPN: Detecting and editing privacy neurons in pretrained language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 2875–2886, 2023. doi: 10.18653/v1/2023.emnlp-main.174.

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*, 18:4732–4746, 2023. doi: 10.1109/TIFS.2023.3297905.

Y. Zhao, P. Wang, H. Qi, J. Huang, Z. Wei, and Q. Zhang. Federated unlearning with momentum degradation. *IEEE Internet of Things Journal*, 11(5):8860–8870, 2024. doi: 10.1109/JIOT.2023.3321594.

Z. Zhong, W. Bao, J. Wang, S. Zhang, J. Zhou, L. Lyu, and W. Y. B. Lim. Unlearning through knowledge overwriting: Reversible federated unlearning via selective sparse adapter. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 30661–30670, 2025. doi: 10.1109/CVPR52734.2025.02852.

X. Zhu, G. Li, and W. Hu. Heterogeneous federated knowledge graph embedding learning and unlearning. In *Proceedings of the ACM Web Conference 2023*, pp. 2444–2454. ACM, 2023. doi: 10.1145/3543507.3583305.

A FEASIBILITY ANALYSIS OF FEDCCCU

This appendix provides a feasibility analysis for our proposed contribution-aware unlearning method, FedCCCU. The core hypothesis of our method is that a neuron's contribution (i.e., attribution score) to a specific concept, such as 'class 0', varies significantly depending on the client's data distribution.

To validate this hypothesis, we design a cross-domain experiment utilizing a CNN model trained in a **CD-NonIID** scenario. Specifically, this scenario involves two clients holding the D_{MNIST} and D_{SVHN} datasets, respectively. While both datasets share a consistent task objective (e.g., recognizing digits 0–9), their underlying feature distributions are markedly different. We subsequently employ attribution methods to separately calculate the sensitive neurons and their contribution scores that the model relies upon for recognizing D_{MNIST} and D_{SVHN}.

A.1 POSITIONAL OVERLAP ANALYSIS

We analyze the limitations of unlearning methods based solely on neuron position. We identify the indices of the Top-100 sensitive neurons with the highest contribution scores for 'class 0' in both D_{MNIST} and D_{SVHN}. Figure 6 shows that the sets share 63 neurons out of 137 in their union, giving a Jaccard overlap of approximately 46%. Removing neurons based only on their importance to D_{MNIST} may therefore affect shared neurons that also support D_{SVHN}.

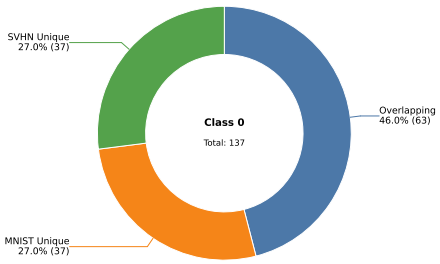

Figure 6: Positional overlap of the Top-100 sensitive neurons for 'class 0' between D_{MNIST} and D_{SVHN}. The sets share 63 neurons, giving a Jaccard overlap of $63/137 \approx 46\%$.

A.2 CONTRIBUTION SCORE ANALYSIS

Next, we further analyze these shared sensitive neurons to investigate whether they can be differentiated by their contribution scores. Figure 7 visualizes these neurons in a 3D plot, where the X and Y axes represent their physical location (layer and index), and the Z-axis represents the contribution score to a specific dataset. The plot shows that even for the exact same position, the contribution scores (Z-axis height) differ significantly across domains.

However, merely observing that contribution scores differ is insufficient. The critical challenge lies in the *Utility Conflict*: a neuron might contribute highly to the forgetting client (e.g., D_{MNIST}) while simultaneously being crucial for the same class in other retaining clients (e.g., D_{SVHN}).

Therefore, we employ the **Utility Conflict Ratio** strategy (referred to as the Dominant Neuron method). The objective is to prioritize neurons with lower estimated retaining client sensitivity relative to requester sensitivity. Editing these candidates aims to suppress requester target predictions while reducing collateral loss. The experimental results in Section 5 evaluate this tradeoff.

B SUPPLEMENTARY EXPERIMENTS: COMPARISON WITH FUSED

In this section, we conduct a detailed experimental comparison between our method FedCCCU and the recently proposed FUSED (Zhong et al., 2025) method. We evaluate FUSED using the same CD-NonIID experimental protocol described in the main paper.

B.1 CLIENT-WISE ACCURACY ANALYSIS

Figure 8 shows the per-client accuracy comparison for FUSED under CD-NonIID. Across all three tasks, the requester C1 shows a large target accuracy drop after unlearning, indicating target suppression. Retaining clients also lose target accuracy, with several falling to 0%. These results show that the suppression extends beyond the requester and harms retaining client utility.

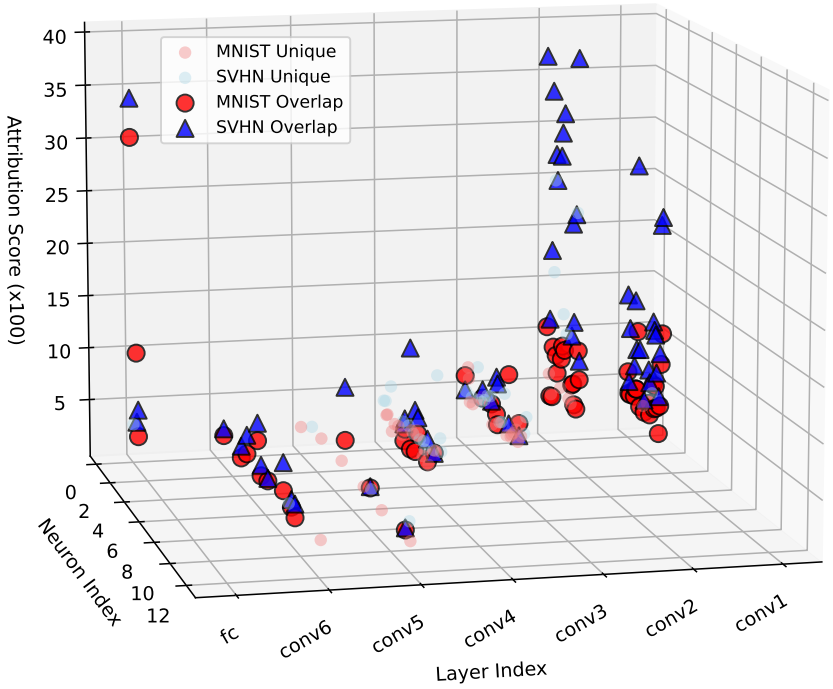

Figure 7: 3D visualization of the sensitive neurons. The X and Y axes represent the neuron's physical location, while the Z-axis plots its contribution score. For each shared (X,Y) location, the solid red dot shows the score for D_{MNIST} and the solid blue triangle shows the score for D_{SVHN}. The divergence in Z-heights demonstrates that contribution scores differ significantly even for positionally-overlapped neurons.

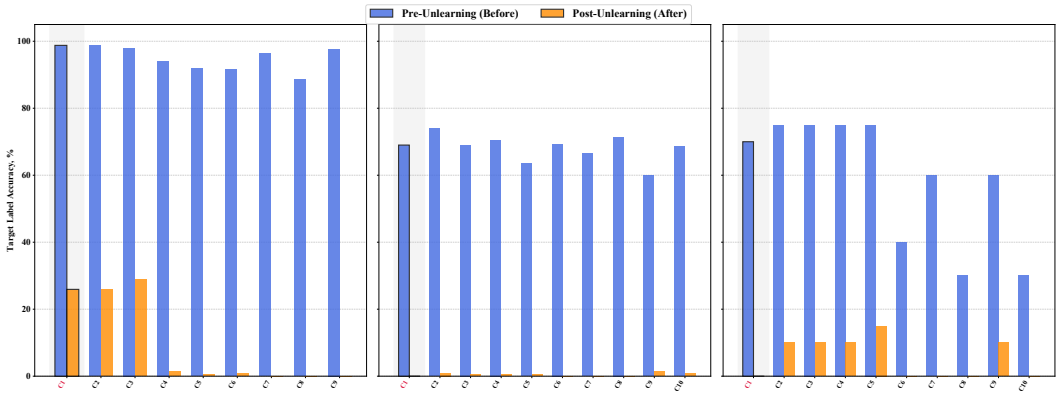

Figure 8: Comparison of Target Label Accuracy on FUSED before and after unlearning. Each column corresponds to a different task setting: (Left) HDR Task (9 clients), (Middle) IR-9 Task (10 clients), and (Right) IR-65 Task (10 clients).

B.2 Heatmap Analysis of Collateral Damage

Figure 9 presents the accuracy drop ($\Delta = \text{acc}_{\text{before}} - \text{acc}_{\text{after}}$) heatmaps for FUSED and `FedCCCU` on the IR-65 task. At the target, Label 0 on Client 1 (C1, L0), FUSED produces stronger suppression (deeper green) than `FedCCCU`, alongside larger retaining client losses.

FUSED also reduces accuracy on Label 0 for retaining clients C2–C10, as indicated by the green cells in the same column. The corresponding changes for `FedCCCU` are smaller. Across the displayed labels, FUSED also produces more accuracy changes outside Label 0, including losses on other client label pairs.

This comparison illustrates the tension between requester target suppression and retaining utility under CD-NonIID. `FedCCCU` reduces collateral loss in this setting, with weaker requester suppression than FUSED.

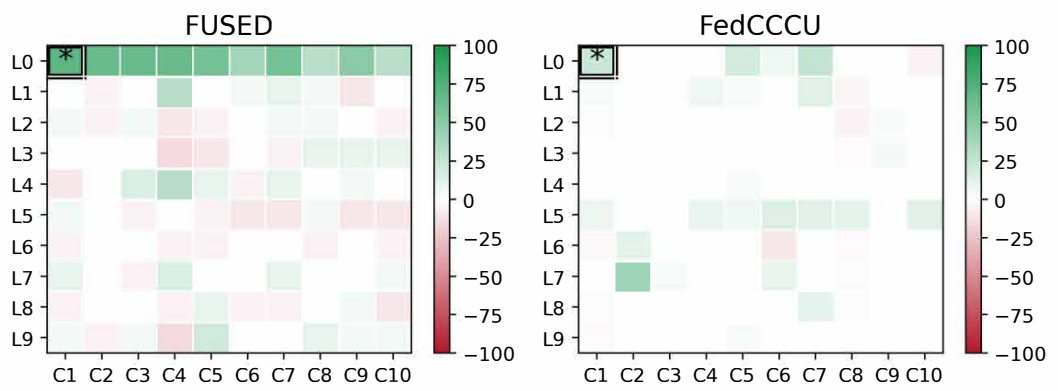

Figure 9: Accuracy drops ($\Delta = \text{acc}_{\text{before}} - \text{acc}_{\text{after}}$) in % for FUSED (left) and `FedCCCU` (right) on IR-65. We show only L0–L9 because space is limited for the 65 classes. The target is (Client 1, Label 0). Green indicates accuracy loss, red indicates gain, and white indicates stability.

C Additional Privacy Evaluation Results

To assess privacy leakage under the evaluated attack, we conducted Membership Inference Attacks (MIA) on the unlearned target class. Table 7 reports the Attack Success Rate (ASR) across all methods.

Table 7: Evaluation of privacy leakage via Membership Inference Attack (MIA). We report the Attack Success Rate (ASR, %) on the unlearned target class. Values closer to 50% (random guessing) indicate lower leakage under this attack.

Task	Attack Success Rate (ASR) (%)								
	Retrain-PI	**Amnesiac-R**	**Amnesiac**	**Ferrari**	**FedCDP**	**FedAU**	**FCU**	**DEPN**	**`FedCCCU`**
HDR	78.33	76.84	50.89	61.32	61.84	51.66	52.54	51.49	50.61
IR-9	72.25	74.72	52.36	59.33	56.13	50.85	52.56	52.24	51.37
IR-65	67.49	65.32	50.89	50.32	52.91	52.87	53.02	51.67	52.53

Sensitivity summary size. During preparation before the request, each client uploads a Top-N list of neuron indices and sensitivity scores for each class. We count each convolutional output channel and fully connected output unit as one neuron, following the sensitivity calculation in the main text. Table 8 reports the fraction of neurons included in each list. Each list covers less than 1% of the model's neurons; this fraction describes summary size and does not establish a privacy guarantee.

Table 8: Size of the neuron sensitivity summaries shared by each client.

Setting	Model	Total neurons	Shared neurons N	Shared ratio
IR-9	ResNet18	4809	30	0.624%
IR-65	ResNet32	10625	20	0.188%

D Additional Related Work

FedEraser (Liu et al., 2021) uses stored historical gradients to approximate retraining. Amnesiac (Graves et al., 2021) stores update history and removes selected updates when an unlearning request arrives. Lin et al. (2024) shard clients and use coding to reduce retraining cost, while DeepObliviate (He et al., 2021), Liu et al. (2022b), and SIFU (Fraboni et al., 2024) further reduce the cost of retraining based unlearning. These methods improve efficiency compared with full retraining, but many of them still require extra work from clients that did not request unlearning.

Other studies edit the current model instead of replaying training. Knowledge distillation and soft label recovery methods (Wu et al., 2022; Zhao et al., 2024; Zhu et al., 2023) reduce the model response to the target data. Gradient ascent and feature filtering methods (Jang et al., 2023; Gu et al., 2024a; Wang et al., 2024) change gradients or features related to the target data. Pruning based methods, such as RevFRF (Liu et al., 2022a), FedCDP (Wang et al., 2022), and DEPN (Wu et al., 2023), remove weights or neurons linked to the target class and then recover model utility. FCU (Deng et al., 2024) combines contrastive unlearning with memory preservation, and FedAU (Gu et al., 2024b) uses an auxiliary unlearning module for fast updates. FUSED (Zhong et al., 2025) trains sparse adapters to overwrite target knowledge, while FedOSD (Pan et al., 2025) uses gradient projection to reduce conflict during unlearning. VeriFi (Gao et al., 2024) adds verifiability to FU, but it also increases the system cost.

FL research outside unlearning has studied fairness, resource allocation, and data heterogeneity. FedDyn (Acar et al., 2021) and FedBN (Li et al., 2021) improve learning under heterogeneous clients. LEAF (Caldas et al., 2018) provides benchmarks with heterogeneous federated data. These ideas are useful, but they have not been fully used in FU evaluation. This gap is the reason we test unlearning in CD-NonIID settings and report its effect on retaining clients.

Table 9: Overview of the Dataset

Task	Dataset	Classes	Train	Test
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

E Data Details

Table 9 summarizes the basic statistics of the six datasets used in our `RWFUBench`. Note that for the cross-domain tasks, we filter these datasets to retain only the overlapping classes as described in the main text. Experiments used a workstation with two NVIDIA RTX 4090 GPUs (22GB each).

F Evaluation of the Global Retraining Baseline

We evaluate global retraining on the retained data as a reference for removing the requester's training data influence. This procedure requires retaining clients to repeat training and therefore violates Process Independence. Table 10 shows that target accuracy remains high after retraining. This is compatible with generalization from target class data retained by other clients and does not establish failure to remove the deleted data's influence. The comparison distinguishes data deletion from requester target suppression and shows the additional participation required by global retraining.

Table 10: Comparison of global and target accuracy between Retrain-All and Retrain-PI before and after unlearning.

Dataset	Method	Pre (%)		Post (%)	
		Global	Target	Global	Target
HDR	Retrain-All	96.82	98.42	95.91	96.12
	Retrain-PI	96.82	99.65	95.30	99.82
IR-9	Retrain-All	90.77	94.33	89.97	85.50
	Retrain-PI	90.77	94.33	88.46	82.77
IR-65	Retrain-All	72.99	62.86	70.23	47.93
	Retrain-PI	72.99	62.86	69.34	45.71

G Client-Level Unlearning under CD-NonIID

We extend the benchmark to client-level unlearning to examine Cross-Domain Knowledge Replenishment and retaining client utility loss beyond class-level deletion. We use the same CD-NonIID tasks, model architectures, and eight FU methods. C1 requests the removal of all its local training samples, while the remaining clients retain their data. Table 11 reports the previously defined client accuracy A_k before and after unlearning. Methods requiring historical training states use checkpoints from their pipelines. Each method's three runs start from the same pre-unlearning checkpoint.

Across all three tasks, Retrain-PI keeps C1 accuracy close to its pre-unlearning level after all C1 training samples are removed. This pattern is consistent with Cross-Domain Knowledge Replenishment under shared labels and cross-domain features. Results for individual clients also reveal retaining

Table 11: Client-level unlearning across CD-NonIID tasks. The bold example **98.93 / 98.33 (0.23)** denotes 98.93% accuracy before unlearning, 98.33% mean accuracy after unlearning, and a sample standard deviation of 0.23% across three runs. C1 is the requesting client.

Client	Retrain-PI	Amnesiac-R	Amnesiac	Ferrari	FedCDP	FedAU	FCU	DEPN
				(a) HDR				
C1	98.93 / 98.33 (0.23)	98.93 / 25.14 (2.02)	99.29 / 97.75 (0.27)	98.84 / 98.89 (0.01)	99.31 / 99.22 (0.03)	98.97 / 0.22 (0.03)	98.24 / 98.39 (0.13)	98.93 / 98.97 (0.00)
C2	99.06 / 98.42 (0.14)	99.06 / 24.14 (1.85)	99.34 / 97.76 (0.10)	99.06 / 98.99 (0.00)	99.43 / 99.29 (0.06)	99.03 / 0.24 (0.06)	98.15 / 98.37 (0.06)	99.06 / 98.96 (0.00)
C3	98.77 / 98.25 (0.19)	98.77 / 24.67 (1.96)	99.55 / 97.55 (0.33)	99.24 / 98.59 (0.05)	99.22 / 99.07 (0.14)	98.66 / 0.04 (0.06)	98.66 / 98.81 (0.19)	98.77 / 98.89 (0.00)
C4	95.22 / 95.43 (0.13)	95.22 / 48.12 (3.92)	93.90 / 95.36 (0.18)	92.84 / 93.65 (0.04)	99.99 / 95.42 (0.06)	93.71 / 65.92 (1.85)	86.91 / 89.07 (0.75)	95.22 / 94.51 (0.00)
C5	95.81 / 95.81 (0.02)	95.81 / 47.40 (3.32)	94.34 / 95.88 (0.09)	93.13 / 94.38 (0.01)	94.46 / 95.96 (0.03)	93.66 / 66.05 (1.88)	86.70 / 89.67 (0.56)	95.81 / 95.22 (0.00)
C6	95.72 / 95.78 (0.19)	95.72 / 47.55 (3.19)	94.33 / 95.92 (0.00)	93.26 / 94.33 (0.02)	94.47 / 96.13 (0.06)	93.95 / 65.91 (2.17)	86.79 / 89.44 (0.61)	95.72 / 95.18 (0.00)
C7	73.91 / 86.71 (2.24)	73.91 / 17.87 (4.48)	95.65 / 54.11 (6.38)	85.38 / 64.98 (0.68)	94.93 / 91.55 (1.23)	97.83 / 14.25 (2.39)	95.65 / 96.14 (1.37)	73.91 / 67.39 (0.00)
C8	72.60 / 83.73 (2.08)	72.60 / 15.85 (2.83)	94.13 / 54.43 (5.35)	89.30 / 61.81 (0.35)	93.55 / 89.37 (0.83)	95.34 / 14.56 (1.49)	94.86 / 95.30 (0.46)	72.60 / 65.34 (0.00)
C9	72.69 / 85.65 (0.38)	72.69 / 14.35 (2.30)	95.37 / 51.70 (8.50)	91.82 / 59.10 (0.58)	95.37 / 92.44 (0.87)	95.37 / 14.81 (2.30)	98.83 / 96.91 (0.44)	72.69 / 63.89 (0.00)

Table 11: Client-level unlearning across CD-NonIID tasks: (b) IR-9 (continued).

Client	Retrain-PI	Amnesiac-R	Amnesiac	Ferrari	FedCDP	FedAU	FCU	DEPN
				(b) IR-9				
C1	94.04 / 92.29 (0.44)	94.04 / 28.50 (1.98)	85.89 / 38.07 (9.63)	82.83 / 51.19 (0.63)	86.17 / 71.30 (8.15)	85.70 / 61.85 (3.76)	88.23 / 86.65 (1.32)	94.04 / 85.48 (0.00)
C2	92.35 / 88.89 (1.57)	92.35 / 30.24 (1.36)	84.78 / 37.89 (10.75)	83.24 / 47.36 (0.26)	84.28 / 69.95 (7.82)	85.79 / 56.47 (4.64)	87.98 / 86.74 (1.05)	92.35 / 86.34 (0.00)
C3	94.76 / 92.98 (0.43)	94.76 / 27.65 (2.06)	84.78 / 40.01 (10.32)	85.02 / 51.86 (0.44)	85.00 / 74.00 (7.35)	85.97 / 59.90 (4.46)	88.88 / 85.80 (1.45)	94.76 / 87.37 (0.00)
C4	95.42 / 93.61 (0.34)	95.42 / 29.23 (1.23)	85.44 / 38.08 (10.21)	83.26 / 50.60 (0.47)	85.61 / 71.15 (8.55)	86.43 / 61.30 (4.54)	87.76 / 86.46 (1.36)	95.42 / 86.85 (0.00)
C5	94.67 / 93.44 (0.11)	94.67 / 28.65 (1.34)	86.06 / 37.87 (10.25)	82.63 / 50.42 (0.39)	85.67 / 71.21 (8.12)	87.41 / 60.82 (4.55)	86.86 / 87.13 (1.32)	94.67 / 86.68 (0.00)
C6	85.99 / 85.85 (0.61)	85.99 / 28.18 (1.23)	72.46 / 38.04 (6.90)	73.91 / 57.95 (0.25)	73.33 / 61.37 (8.36)	75.32 / 54.08 (2.23)	82.25 / 81.85 (1.21)	85.99 / 81.97 (0.00)
C7	85.08 / 84.12 (0.73)	85.08 / 27.37 (1.55)	70.87 / 37.06 (7.14)	72.69 / 58.60 (0.47)	70.87 / 60.20 (7.96)	73.90 / 53.88 (2.83)	80.44 / 81.64 (1.04)	85.08 / 80.52 (0.00)
C8	94.54 / 83.68 (0.12)	94.54 / 27.06 (1.80)	71.90 / 40.03 (6.21)	70.43 / 61.86 (0.73)	72.06 / 61.43 (7.56)	71.65 / 51.37 (3.19)	79.12 / 81.72 (1.32)	94.54 / 81.44 (0.00)
C9	83.33 / 81.44 (0.51)	83.33 / 28.86 (1.52)	71.35 / 40.24 (5.87)	67.26 / 60.30 (0.19)	71.35 / 58.13 (7.06)	68.70 / 48.24 (2.76)	77.26 / 81.85 (0.91)	83.33 / 80.08 (0.00)
C10	86.41 / 85.69 (1.18)	86.41 / 28.24 (1.22)	71.83 / 37.96 (7.32)	73.79 / 57.39 (0.22)	72.30 / 60.12 (7.97)	74.58 / 54.00 (2.21)	81.56 / 81.35 (0.56)	86.41 / 81.28 (0.00)

client utility loss. On HDR, Amnesiac, Ferrari, and DEPN leave C1 nearly unchanged while reducing accuracy for C7–C9. Several methods produce broader retaining client drops on the IR tasks. These effects can be hidden by global accuracy or C1 accuracy alone. Together, the results show that the replenishment and retaining client utility loss patterns found in the main experiments also appear under client-level unlearning.

H Target-Class Sensitivity Analysis

To evaluate sensitivity to the target class, we keep the requesting client, initial checkpoint, and task settings unchanged and vary only c. We evaluate three target classes on each task to test whether the editing effect remains concentrated on the requesting client when the target class changes.

Across all tested classes in Table 12, the accuracy change is larger on C_1 than on retaining clients, while global accuracy changes less. Changing c alters the sensitive neurons identified by attribution, and the Utility Conflict Ratio prioritizes candidates with lower estimated relative conflict. This pattern persists without class-specific tuning and is not limited to label 0.

Table 11: Client-level unlearning across CD-NonIID tasks: (c) IR-65 (continued).

Client	Retrain-PI	Amnesiac-R	Amnesiac	Ferrari	FedCDP	FedAU	FCU	DEPN
				(c) IR-65				
C1	66.54 / 64.39 (1.34)	66.54 / 17.02 (11.21)	56.00 / 62.59 (0.48)	60.52 / 12.30 (0.64)	59.00 / 68.49 (0.33)	58.96 / 66.11 (2.95)	66.77 / 65.38 (0.50)	46.64 / 55.46 (0.00)
C2	68.62 / 63.12 (2.32)	68.62 / 18.83 (11.43)	58.15 / 61.57 (1.10)	59.28 / 11.52 (0.31)	61.15 / 67.44 (1.24)	61.79 / 65.67 (1.91)	68.92 / 68.77 (0.44)	46.69 / 52.16 (0.00)
C3	66.77 / 63.18 (2.54)	66.77 / 15.67 (10.57)	58.08 / 61.44 (1.53)	56.88 / 18.16 (0.93)	59.54 / 65.92 (1.27)	61.94 / 67.16 (3.81)	68.85 / 67.21 (0.49)	46.15 / 51.49 (0.00)
C4	69.15 / 67.26 (0.80)	69.15 / 15.51 (10.57)	59.62 / 64.80 (0.79)	63.56 / 12.32 (0.18)	60.54 / 71.48 (1.01)	63.03 / 68.68 (1.77)	69.69 / 69.31 (0.70)	48.77 / 56.55 (0.00)
C5	68.77 / 62.69 (1.76)	68.77 / 17.41 (11.27)	60.38 / 61.66 (0.81)	61.19 / 12.96 (0.60)	61.92 / 66.52 (0.47)	59.52 / 65.62 (2.46)	68.92 / 67.51 (0.74)	48.62 / 53.72 (0.00)
C6	72.31 / 72.64 (1.32)	72.31 / 29.04 (19.33)	55.54 / 72.20 (0.62)	58.69 / 20.54 (0.88)	56.77 / 73.62 (0.70)	57.37 / 74.41 (1.72)	60.69 / 59.90 (0.25)	42.31 / 70.39 (0.00)
C7	70.08 / 71.97 (0.98)	70.08 / 27.74 (17.34)	54.00 / 71.53 (0.57)	60.25 / 20.41 (0.37)	54.92 / 73.06 (0.74)	53.59 / 72.40 (1.55)	56.92 / 59.79 (0.19)	42.00 / 68.85 (0.00)
C8	70.54 / 71.66 (0.73)	70.54 / 27.55 (18.00)	56.08 / 71.27 (0.34)	56.67 / 18.23 (0.53)	55.54 / 73.25 (0.30)	56.33 / 72.14 (1.50)	59.54 / 60.28 (0.40)	39.15 / 67.86 (0.00)
C9	71.92 / 73.40 (1.15)	71.92 / 26.77 (18.55)	54.23 / 72.16 (1.52)	56.62 / 20.92 (1.40)	56.85 / 74.29 (1.09)	60.11 / 71.63 (1.33)	58.31 / 60.72 (0.58)	42.23 / 68.62 (0.00)
C10	69.92 / 68.57 (0.86)	69.92 / 26.11 (16.56)	53.46 / 69.85 (0.35)	55.16 / 18.26 (0.77)	54.85 / 71.80 (0.66)	52.25 / 69.48 (1.20)	56.00 / 58.10 (0.49)	38.38 / 65.40 (0.00)

I Impact Statement

This work advances trustworthy federated unlearning by providing a benchmark and analysis framework for evaluating whether federated unlearning methods can satisfy data deletion requirements without imposing cost or utility loss on uninvolved clients. By exposing the gap between Pseudo-NonIID and CD-NonIID evaluations, our findings can help researchers and practitioners design more reliable federated unlearning protocols for privacy compliance and user data rights. We also acknowledge a potential dual-use risk: systematic failure analysis may reveal weaknesses in deployed unlearning pipelines or be used to overstate compliance when evaluation settings are too simplified. To mitigate this risk, we advocate transparent reporting of data assumptions, client participation requirements, retain-client utility metrics, and privacy tests such as membership inference.

Table 12: Target class sensitivity of `FedCCCU`. Each cell reports accuracy before unlearning and the mean $\pm$ sample standard deviation after unlearning across three runs (%). The C1 c column reports target class accuracy on C_1, and Ret. c reports its mean over C_2–C_{10}.

Task	Target class	Global	C1 c	Ret. c
IR-9	airplane	$90.77/82.51 \pm 0.52$	$94.33/19.07 \pm 1.32$	$93.23/71.89 \pm 1.42$
	bird	$90.77/81.88 \pm 1.02$	$91.67/21.83 \pm 0.16$	$90.92/76.59 \pm 1.03$
	ship	$90.77/82.84 \pm 0.89$	$98.17/42.33 \pm 2.75$	$89.33/74.91 \pm 0.75$
IR-65	apple	$69.46/68.92 \pm 0.94$	$62.86/44.16 \pm 1.02$	$71.11/68.61 \pm 2.31$
	crab	$69.46/67.94 \pm 1.56$	$51.73/35.42 \pm 0.67$	$52.22/48.89 \pm 0.22$
	crocodile	$69.46/68.35 \pm 1.47$	$55.41/46.37 \pm 0.43$	$65.56/65.21 \pm 0.43$

J Class-Level Multi-Seed Stability under CD-NonIID

Table 13 reports the complete client-wise values corresponding to Figure 3.

Table 13: Class-level multi-seed target accuracy under CD-NonIID: (a) HDR and (b) IR-9. Each cell reports pre-unlearning accuracy/post-unlearning mean with the sample standard deviation in parentheses across three runs (%). C1 is the requesting client.

Client	Retrain-PI	Amnesiac-R	Amnesiac	Ferrari	FedCDP	FedAU	FCU	DEPN
				(a) HDR				
C1	99.65 / 99.82 (0.17)	99.65 / 98.42 (0.50)	99.82 / 10.68 (0.42)	99.65 / 52.67 (2.19)	99.82 / 59.19 (1.87)	99.65 / 0.00 (0.00)	96.85 / 0.00 (0.00)	99.65 / 1.93 (1.71)
C2	99.68 / 99.68 (0.38)	99.68 / 98.72 (0.27)	100.00 / 9.62 (2.48)	99.69 / 55.62 (1.28)	100.00 / 59.62 (0.26)	100.00 / 0.00 (0.00)	96.79 / 0.00 (0.00)	99.68 / 0.96 (0.28)
C3	100.00 / 100.00 (0.00)	100.00 / 98.97 (0.89)	100.00 / 4.12 (0.84)	100.00 / 58.59 (2.65)	100.00 / 48.45 (2.35)	100.00 / 0.00 (0.00)	98.00 / 0.00 (0.00)	100.00 / 1.03 (0.15)
C4	96.35 / 97.08 (1.05)	96.35 / 91.97 (2.70)	96.35 / 79.56 (2.55)	94.03 / 18.19 (1.79)	96.35 / 94.89 (1.27)	94.16 / 0.00 (0.00)	83.21 / 37.96 (1.84)	96.35 / 2.18 (0.53)
C5	96.67 / 96.44 (0.75)	96.67 / 93.71 (2.22)	96.32 / 78.38 (0.42)	95.72 / 17.96 (2.39)	96.32 / 91.45 (1.29)	94.42 / 0.00 (0.00)	80.29 / 98.00 (2.38)	96.67 / 2.73 (0.23)
C6	97.52 / 97.52 (1.79)	97.52 / 95.95 (0.65)	96.86 / 78.17 (0.11)	96.33 / 17.43 (0.22)	96.73 / 91.50 (0.87)	95.42 / 0.00 (0.00)	79.87 / 97.78 (2.07)	97.52 / 1.96 (1.89)
C7	53.57 / 100.00 (0.00)	53.57 / 89.29 (2.86)	100.00 / 0.00 (0.00)	78.57 / 0.00 (0.00)	100.00 / 85.71 (1.11)	100.00 / 0.00 (0.00)	96.43 / 75.00 (2.65)	53.57 / 0.00 (0.00)
C8	67.24 / 95.82 (2.66)	67.24 / 94.77 (1.07)	96.52 / 0.00 (0.00)	87.10 / 0.00 (0.00)	96.17 / 82.93 (1.56)	96.17 / 0.00 (0.00)	97.56 / 86.41 (1.13)	67.24 / 0.00 (0.00)
C9	75.00 / 95.45 (0.70)	75.00 / 93.18 (0.92)	95.45 / 0.00 (0.00)	83.85 / 0.00 (0.00)	95.45 / 93.18 (0.39)	95.45 / 0.00 (0.00)	97.73 / 93.19 (2.24)	75.00 / 0.00 (0.00)

Table 13: Class-level multi-seed target accuracy under CD-NonIID: (b) IR-9 (continued).

Client	Retrain-PI	Amnesiac-R	Amnesiac	Ferrari	FedCDP	FedAU	FCU	DEPN
				(b) IR-9				
C1	94.33 / 82.77 (1.90)	94.33 / 83.67 (2.37)	87.00 / 0.00 (0.00)	83.39 / 54.04 (0.73)	85.00 / 35.50 (1.30)	82.31 / 0.00 (0.00)	85.26 / 0.91 (1.17)	94.33 / 7.94 (2.02)
C2	100.00 / 77.78 (0.44)	100.00 / 77.78 (1.59)	83.00 / 0.00 (0.00)	77.78 / 50.00 (1.78)	82.00 / 40.00 (1.37)	88.89 / 0.00 (0.00)	83.33 / 0.00 (0.00)	100.00 / 5.56 (0.18)
C3	94.05 / 83.78 (1.63)	94.05 / 84.86 (2.18)	88.50 / 0.00 (0.00)	84.98 / 60.81 (0.52)	87.50 / 37.00 (1.55)	84.32 / 1.62 (0.79)	85.41 / 0.00 (0.00)	94.05 / 9.73 (0.45)
C4	94.59 / 83.78 (2.21)	94.59 / 85.41 (1.10)	83.50 / 0.00 (0.00)	86.62 / 60.52 (2.85)	82.50 / 40.00 (0.90)	86.49 / 0.00 (0.00)	85.95 / 2.16 (1.31)	94.59 / 8.65 (0.19)
C5	97.08 / 85.96 (0.28)	97.08 / 83.04 (0.71)	87.00 / 0.00 (0.00)	82.39 / 50.90 (0.29)	84.50 / 41.50 (1.97)	86.55 / 1.17 (1.09)	87.13 / 1.17 (1.41)	97.08 / 8.19 (0.50)
C6	90.32 / 83.87 (0.08)	90.32 / 80.06 (1.90)	86.43 / 0.00 (0.00)	86.92 / 66.57 (2.17)	86.43 / 45.00 (1.91)	90.03 / 0.00 (0.00)	88.56 / 0.00 (0.00)	90.32 / 17.01 (0.46)
C7	92.41 / 79.11 (1.50)	92.41 / 83.54 (2.77)	89.29 / 1.43 (0.96)	87.29 / 60.67 (2.71)	89.29 / 44.29 (0.58)	88.61 / 0.00 (0.00)	90.51 / 0.00 (0.00)	92.41 / 17.72 (1.10)
C8	90.74 / 77.78 (1.63)	90.74 / 85.19 (1.53)	87.14 / 0.71 (0.42)	81.39 / 59.94 (0.37)	86.43 / 45.71 (1.89)	74.07 / 0.00 (0.00)	85.19 / 0.00 (0.00)	90.74 / 14.81 (1.16)
C9	100.00 / 73.91 (0.76)	100.00 / 73.91 (2.40)	80.71 / 0.71 (0.50)	78.26 / 56.52 (0.29)	79.29 / 39.29 (1.26)	82.61 / 0.00 (0.00)	95.65 / 0.00 (0.00)	100.00 / 17.39 (2.85)
C10	93.55 / 79.03 (0.67)	93.55 / 84.48 (1.24)	89.29 / 0.00 (0.00)	91.18 / 69.64 (0.71)	88.57 / 46.43 (0.98)	87.90 / 0.00 (0.00)	89.52 / 0.81 (1.16)	93.55 / 16.94 (0.40)

Table 13: Class-level multi-seed target accuracy under CD-NonIID: (c) IR-65 (continued).

Client	Retrain-PI	Amnesiac-R	Amnesiac	Ferrari	FedCDP	FedAU	FCU	DEPN
				(c) IR-65				
C1	62.86 / 45.71 (0.45)	62.86 / 51.43 (2.03)	75.00 / 0.00 (0.00)	70.83 / 0.00 (0.00)	75.00 / 35.00 (0.59)	68.57 / 0.00 (0.00)	75.00 / 0.00 (0.00)	62.86 / 0.00 (0.00)
C2	57.14 / 57.14 (2.23)	57.14 / 71.43 (1.31)	85.00 / 0.00 (0.00)	85.71 / 0.00 (0.00)	85.00 / 40.00 (1.80)	85.71 / 0.00 (0.00)	95.00 / 10.00 (0.95)	57.14 / 0.00 (0.00)
C3	50.00 / 50.00 (2.14)	50.00 / 50.00 (1.47)	90.00 / 5.00 (2.89)	50.00 / 0.00 (0.00)	90.00 / 35.00 (2.87)	100.00 / 0.00 (0.00)	100.00 / 0.00 (0.00)	50.00 / 0.00 (0.00)
C4	53.85 / 38.46 (1.70)	53.85 / 53.85 (0.43)	85.00 / 10.00 (2.76)	61.54 / 0.00 (0.00)	70.00 / 35.00 (1.98)	76.92 / 0.00 (0.00)	95.00 / 5.00 (0.32)	53.85 / 0.00 (0.00)
C5	76.74 / 62.79 (0.45)	76.74 / 69.77 (0.38)	100.00 / 5.00 (1.73)	76.99 / 0.00 (0.00)	80.00 / 45.00 (0.22)	72.09 / 0.00 (0.00)	95.00 / 5.00 (1.60)	76.74 / 0.00 (0.00)
C6	60.00 / 73.33 (2.36)	60.00 / 73.33 (2.91)	75.00 / 5.00 (0.76)	53.33 / 26.67 (1.91)	65.00 / 55.00 (2.35)	66.67 / 0.00 (0.00)	60.00 / 25.00 (2.58)	60.00 / 0.00 (0.00)
C7	75.00 / 50.00 (2.27)	75.00 / 75.00 (1.18)	75.00 / 10.00 (1.49)	50.00 / 0.00 (0.00)	70.00 / 60.00 (0.90)	50.00 / 0.00 (0.00)	75.00 / 20.00 (0.41)	75.00 / 0.00 (0.00)
C8	80.70 / 78.95 (1.43)	80.70 / 78.95 (2.73)	65.00 / 15.00 (1.53)	67.25 / 7.12 (1.57)	50.00 / 65.00 (2.47)	70.18 / 0.00 (0.00)	65.00 / 5.00 (2.37)	80.70 / 0.00 (0.00)
C9	33.33 / 66.67 (2.53)	33.33 / 66.67 (2.05)	85.00 / 10.00 (0.70)	33.33 / 0.00 (0.00)	80.00 / 65.00 (0.49)	66.67 / 0.00 (0.00)	55.00 / 5.00 (1.99)	33.33 / 0.00 (0.00)
C10	71.43 / 61.90 (2.76)	71.43 / 61.90 (0.99)	75.00 / 10.00 (1.38)	42.86 / 4.76 (1.82)	75.00 / 55.00 (1.69)	42.86 / 0.00 (0.00)	60.00 / 10.00 (2.24)	71.43 / 0.00 (0.00)

K Global Accuracy Drop and Execution Time

Figure 10 reports the global accuracy drop (Δ), which averages changes over clients and labels. Retrain-PI shows a small global drop alongside high requester target accuracy. This pattern is consistent with retaining clients supplying features related to the target class during aggregation. Several lightweight methods produce stronger target suppression, but their parameter edits can also affect features used by retaining clients. These results illustrate the tension between requester suppression and retaining utility described in Section 4.

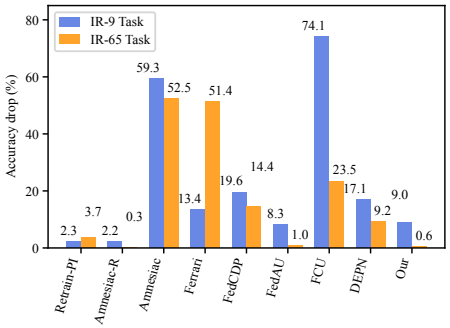

Figure 10: Overall test accuracy drop ($\Delta = \text{acc}_{\text{before}} - \text{acc}_{\text{after}}$) in %. Baseline values are computed from Table 3, and FedCCCU values from Tables 5 and 15.

On IR-65, FedCCCU has a global drop of 0.56%, compared with 1.00% for FedAU, but this aggregate comparison alone cannot judge UI. At the main settings, requester target accuracy drops by 17.70% on IR-65 and 76.33% on IR-9, while retaining target accuracy drops by 2.78% and 22.19%, respectively. These results show a larger effect on the requester in both tasks, with different suppression strengths and remaining collateral loss. Figure 5 complements this aggregate comparison with client wise heatmaps.

Table 14 shows that FedCCCU is faster than retraining based baselines and remains close to practical model editing methods in the tested tasks. This runtime is consistent with the lightweight goal of the PoC. The reported times for FedCCCU are 91, 102, and 138 seconds on HDR, IR-9, and IR-65, respectively, compared with 81, 87, and 116 seconds for DEPN.

Table 14: Execution time in seconds for different methods across the three evaluation tasks.

Method	HDR	IR-9	IR-65
Retrain-PI	841	3122	8821
Amnesiac-R	963	2715	9247
Amnesiac	785	2945	8463
Ferrari	13	51	133
FedCDP	183	428	1381
FedAU	$< 10^{-3}$	$< 10^{-3}$	$< 10^{-3}$
FCU	52	242	781
DEPN	81	87	116
FedCCCU	91	102	138

L Parameter Sensitivity on IR-9

Table 15 examines candidate selection and edit strength on IR-9. The N sweep shows that candidate composition matters at a fixed edit size. At $N = 30$, the main setting $P = 8$ yields target accuracy of 18.00% on C1 and 71.04% on retaining clients. Increasing P further suppresses requester predictions but reduces retaining client accuracy. This comparison shows how the editing budget balances requester suppression against preserving shared target knowledge.

Table 15: Parameter sensitivity of FedCCCU on IR-9. Accuracy values are in %. The N sweep fixes $P = 8$, and the P sweep fixes $N = 30$. Ret. L0 is the mean target class accuracy of C2–C10. Bold values show the main setting.

			Candidate neurons N ($P = 8$)					Pruned neurons P ($N = 30$)					
Task	**Metric**	**Pre**	**20**	**30**	**50**	**70**	**100**	**1**	**2**	**4**	**8**	**16**	**30**
IR-9	Global	90.77	89.18	**81.82**	89.07	89.26	85.73	90.72	90.59	90.01	**81.82**	78.67	20.95
	C1-L0	94.33	89.50	**18.00**	92.62	95.13	86.32	94.58	93.51	94.57	**18.00**	16.37	0.53
	Ret. L0	93.23	88.85	**71.04**	92.37	92.96	88.34	92.79	93.00	93.47	**71.04**	16.74	1.15