DERE: CLASS SUPPRESSION IN CROSS DOMAIN FEDERATED UNLEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated unlearning (FU) aims to remove requested training effects from a federated model without retraining from scratch. Some FU requests go beyond excluding data from training and require the resulting global model to have low accuracy on the requested data while preserving retained knowledge. We extend this goal to cross domain FU, a more practical setting where clients may have different input distributions. After unlearning, the global model should suppress the requested classes on the forget client's domain while preserving the same classes on other domains and performance on the remaining data. In this setting, class information from other clients and domain information retained by the forget client may combine during aggregation and restore the target behavior. We refer to this recovery path as *Update Reintroduction* (UR). Stronger deletion can reduce UR but may also hurt retained accuracy. To address this tension, we propose **DeRe** (De-Reintroduction) with three stages. Probe Based Mask Selection identifies the edit region using relative deletion responses. Projected Class Level Unlearning coordinates deletion and retained updates at the client and server. Mask Constrained Recovery restores utility outside the selected mask without overwriting the selected edits. Experiments on Digit5, DomainNet, and PACS show that DeRe improves the balance between target suppression and retained accuracy for both single and multi class requests. Code is available at DeRe.

1 INTRODUCTION

Federated learning (FL) trains a shared model through client updates while keeping raw data local (Wu et al., 2024; Bai et al., 2025). Federated unlearning (FU) handles later requests to remove selected training effects under local data and communication constraints (Liu et al., 2024; Romandini et al., 2025). In practice, clients can also have different input distributions. Medical images from different hospitals may vary because of scanner settings, acquisition conditions, and patient populations (Guan et al., 2024). Such cross-domain shift has been studied in FL (Li et al., 2021), and recent FU work considers client removal under cross-domain settings (Nguyen et al., 2026).

Some FU studies already consider removing one class from one client while the same class remains on other clients (Li et al., 2025b; Qiu et al., 2026). A hospital may request data removal following a privacy policy change, with unlearning evaluated by low accuracy on the requested data and preserved accuracy on retained data (Ong & Chan, 2025). We extend this request to cross-domain FU. Given (f, c), the global model should suppress class c on the forget client's domain while preserving the same class on other clients and the remaining classes on f.

This setting is difficult because both class and domain information must be retained. Non-forget clients continue learning class c, while the forget client's remaining classes carry information about domain f. Aggregation can combine these sources and support recognition of class c on domain f even when the requested samples are excluded. We refer to this possible recovery path as *Update Reintroduction* (UR), illustrated in Figure 1. Stronger deletion updates may reduce this effect but can also lower retained accuracy.

Figure 2 tests whether these retained sources support target recognition during retraining from scratch on Digit5, DomainNet, and PACS. Curve D excludes the forget client and measures recognition transferred from other domains. Curve B also includes the forget client's remaining classes and reaches higher accuracy on forget data (FA). FA stays at 0% when only the forget client trains after

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.

removing class c in Curve A, or when all clients remove class c in Curve C. The increase from D to B is consistent with class information from other domains and domain information from the forget client's remaining classes jointly supporting target recognition.

Existing FU methods reduce retraining cost through update reuse and model reconstruction (Liu et al., 2021; 2022; Li et al., 2025a; Fraboni et al., 2024), or edit models through distillation, pruning, and update adjustment (Wu et al., 2022; Wang et al., 2022; Pan et al., 2025; Zhong et al., 2025). Class-wise FU considers removing a class from one client (Li et al., 2025b; Qiu et al., 2026), while cross-domain FU considers client removal under domain shift (Nguyen et al., 2026). Our setting combines these two cases while keeping the target class on other clients. The interaction between retained class information and retained domain information is not the focus of these methods.

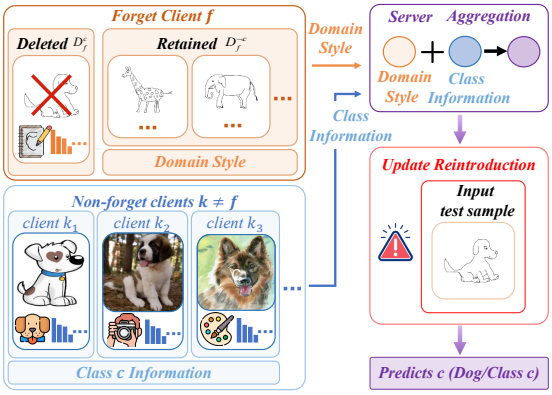

Figure 1: Possible sources of Update Reintroduction in retained data.

We propose DeRe (De-Reintroduction) with three stages. Probe Based Mask Selection (PBMS) selects neurons with stronger deletion responses relative to ordinary probe responses and defines the edit region. Projected Class Level Unlearning (PCLU) coordinates deletion and retained updates at the forget client and server to limit edit size and reduce update cancellation. Mask Constrained Recovery (MCR) restores retained utility outside the selected region, so recovery does not directly overwrite the selected edits. Non-forget clients receive no requested label and continue ordinary training.

We evaluate DeRe on Digit5 (Li et al., 2021), DomainNet (Peng et al., 2019), and PACS (Li et al., 2017) under a shared protocol. In the main single class experiments, DeRe reaches zero FA on all three datasets and retains higher overall accuracy than the compared methods with the same FA. Multi class experiments show the same suppression and retention trade-off. Ablations support PBMS for target suppression, show that PCLU provides the main reduction in retained accuracy loss, and show further gains from MCR.

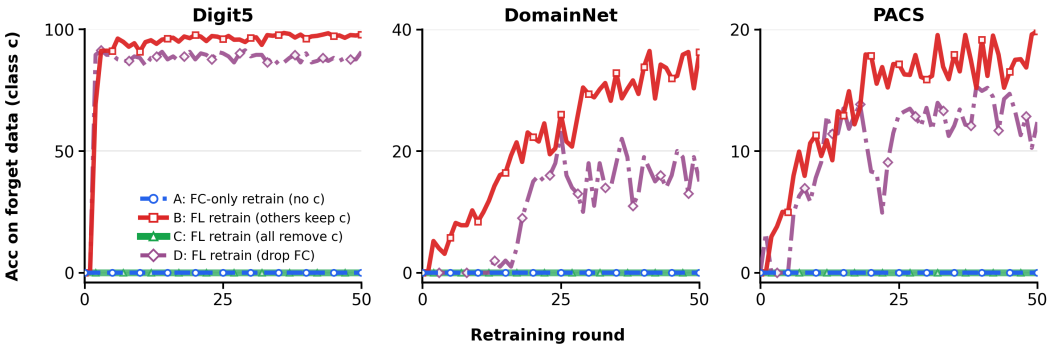

Figure 2: Effect of retained data sources on target recognition. Accuracy on forget data is tracked during retraining on Digit5, DomainNet, and PACS. FC denotes the forget client.

2 Related Work

Federated Unlearning. FU methods reduce removal cost through efficient retraining or direct model editing. FedEraser (Liu et al., 2021) reuses stored updates for client removal, while SIFU (Fraboni et al., 2024) handles sequential client removal. Other approaches use rapid retraining, update histories, or sharding (Liu et al., 2022; Graves et al., 2021; Lin et al., 2024). FedCDP (Wang et al.,

2022) uses pruning for class removal. Other model editing methods use distillation, feature sensitivity, contribution dampening, gradient conflict mitigation, or selective overwriting (Wu et al., 2022; Gu et al., 2024a; Chundawat et al., 2024; Pan et al., 2025; Zhong et al., 2025). Other protocols study communication cost and unlearning guarantees (Tao et al., 2024; Che et al., 2023).

Cross domain federated learning and unlearning. Cross-domain feature shift is different from label imbalance. Clients may share labels but differ in visual style or acquisition conditions. FedBN (Li et al., 2021) shows that local batch normalization can reduce feature shift in FL, and later work studies domain generalization regularization, disentangled domain features, and model merging under cross silo settings (Nguyen et al., 2022; Wang et al., 2025b; Chen et al., 2024). Broader studies on non-IID FL also show that label skew does not cover all feature heterogeneity (Jimenez G. et al., 2024; Jimenez-Gutierrez et al., 2026). FOUL (Nguyen et al., 2026) studies client removal under cross-domain shift. Our setting retains the target class on non-forget clients and the remaining classes on the forget client. These data support retained utility but may also support target recognition through cross-domain transfer.

Knowledge resurfacing. Prior work studies how model updates can restore behavior after unlearning. Łucki et al. (2025) recover suppressed capabilities in language models through adaptive attacks, including fine tuning on unrelated examples. ILU (Wang et al., 2025a) uses invariance regularization to improve resilience to downstream fine tuning. Lethe (Wu et al., 2026) addresses resurfacing during continued federated training through forget and retain update streams. UR concerns a possible recovery path in which target class information from other clients combines with domain information preserved by the forget client's remaining data.

Update conflict and parameter protection. Gradient surgery (Yu et al., 2020) changes directions when task gradients conflict, and Fisher based regularization (Kirkpatrick et al., 2017) protects important parameters. In FU, FedOSD (Pan et al., 2025) and FOUL (Nguyen et al., 2026) use conflict mitigation, while subspace based FU (Li et al., 2025a) and sparse adapter based overwriting reduce the edited parameter space (Zhong et al., 2025). Neuron editing work also shows that target information can be concentrated in a subset of neurons (Wu et al., 2023). These methods provide tools for controlling update directions and edit scope. DeRe uses these tools to coordinate deletion and recovery while other domains retain the target class.

3 Setting and Assumptions

Cross domain setting. We consider K clients indexed by $k \in \{0, \dots, K-1\}$ in an FL system. Client k holds a local training set $\mathcal{D}_k$ sampled from a distribution $\mathcal{P}_k$ over feature space $\mathcal{X}$ and a shared label set $\mathcal{Y}$. All clients use the same labels, but their class conditional feature distributions can differ. We use cross-domain shift to mean that $\mathcal{P}_k(x \mid y)$ may vary across clients for the same label y. The shared model predicts from input features without a client identifier.

Unlearning target. Let θ_{pre} be the global model before unlearning. An unlearning request is denoted by (f, c), where f is the forget client and $c \in \mathcal{Y}$ is the requested class. For class $a \in \mathcal{Y}$, let $\mathcal{D}_k^a = \{(x, y) \in \mathcal{D}_k \mid y = a\}$. The deletion set and retained training set are

$$\mathcal{D}_{\text{del}} = \mathcal{D}_f^c, \qquad \mathcal{D}_{\text{ret}} = \left(\bigcup_{k=0}^{K-1} \mathcal{D}_k \right) \setminus \mathcal{D}_{\text{del}}. \tag{1}$$

Thus, $\mathcal{D}_{\text{ret}}$ includes the non requested data on client f and all data on clients $k \neq f$. In our protocol, each non-forget client contains class c, so $\mathcal{D}_k^c$ for $k \neq f$ remains in $\mathcal{D}_{\text{ret}}$. The request seeks to suppress class c on the forget client's input distribution while preserving its recognition on other clients' domains and accuracy on the remaining classes.

A model retrained from scratch on $\mathcal{D}_{\text{ret}}$ provides the reference for data deletion. It may still recognize class c on client f through cross-domain transfer from retained data. Our behavioral goal additionally requires suppressing that recognition, so we evaluate target suppression and retained utility separately.

Information flow and trust. We assume a trusted server and clients that follow the prescribed protocol. The server receives the request (f, c), but it does not access raw samples or sample level labels from any client. Non-forget clients are not told c and continue ordinary local training. Only the forget client uses local labels to split $\mathcal{D}_f$ into $\mathcal{D}_f^c$ and $\mathcal{D}_f \setminus \mathcal{D}_f^c$. It computes deletion probes and

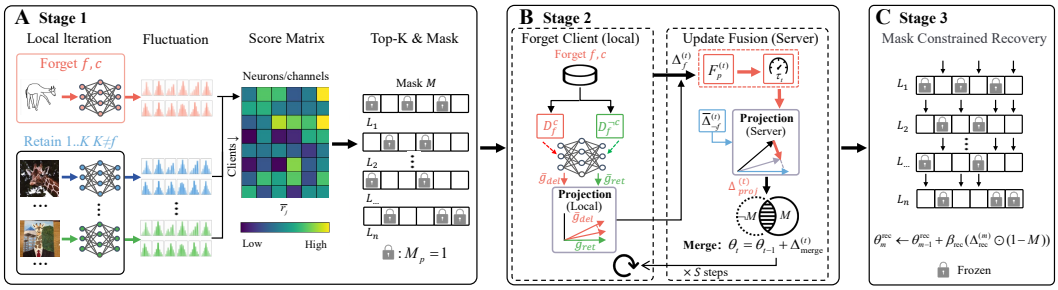

Figure 3: The three stages of **DeRe**. Panel A locates the edit region through PBMS. Panel B coordinates deletion and retained updates through PCLU. Panel C recovers retained utility outside the selected region through MCR.

unlearning updates locally. The server selects the mask, fuses updates, and enforces the recovery mask using uploaded updates and derived statistics.

4 METHOD

4.1 OVERVIEW

DeRe takes the pretrained model θ_{pre} and request (f, c) as input and returns θ_{post} through three stages. PBMS selects the edit region from short probe updates. PCLU coordinates deletion and retained updates within this region, while non-forget clients continue ordinary training. MCR recovers retained utility outside the region to avoid directly overwriting the selected edits. Figure 3 shows these stages.

We set $\theta_0 = \theta_{\text{pre}}$ at the start of PCLU. The binary mask M has the same shape as the model parameters. For a parameter index p, $M_p = 1$ marks a selected parameter that is editable in PCLU and protected during MCR, while $M_p = 0$ marks a parameter outside this region. The operator $\odot$ denotes element wise multiplication, and $\epsilon > 0$ is a small constant used in denominators. For vectors u and v, $\|v\|_2$ denotes the Euclidean norm, $\|v\|_2^2$ its square, and $\langle u, v \rangle$ their inner product.

4.2 PROBE BASED MASK SELECTION (PBMS)

PBMS builds a parameter mask M from short probe updates without updating the global model. The forget client probes deletion on $\mathcal{D}_{\text{del}}$ using the deletion objective in Appendix A. Non-forget clients probe ordinary training on their local data. Let q index the probe runs. Each run starts from a fresh copy of θ_{pre}, and the resulting updates are used only for scoring.

Let $\delta_k^{(q)}$ be the difference between client k's model after probe run q and θ_{pre}. Let j index a neuron or channel, and let $\mathcal{G}_j$ be the set of parameter indices associated with it. For layer ℓ, let $\mathcal{J}_\ell$ be the set of neurons or channels in that layer. For each group, define the forget response and the largest non-forget response as

$$a_{f,j}^{(q)} = \|\delta_f^{(q)}[\mathcal{G}_j]\|_2, \qquad a_{\neg f,j}^{(q)} = \max_{k \neq f} \|\delta_k^{(q)}[\mathcal{G}_j]\|_2. \tag{2}$$

Taking the largest non-forget response accounts for a group that is active on even one retained domain. Let $\mathcal{J}_\ell^{\text{act}} \subseteq \mathcal{J}_\ell$ contain groups with nonzero activity, after the numerical filter in Appendix B.

The server scores each active neuron or channel by averaging the response ratio over the probe runs,

$$\bar{r}_j = \text{Mean}_q \left[\frac{a_{\neg f,j}^{(q)}}{a_{f,j}^{(q)} + \epsilon} \right] \tag{3}$$

A smaller $\bar{r}_j$ gives priority to deletion response relative to ordinary response.

Selection is done separately in each layer to avoid placing most edits in only a few layers. Let B be the edit budget. The server orders active groups by increasing score within each layer, then cycles through layers in a fixed order and takes one group from each nonempty list until B groups are selected or all lists are exhausted. The resulting count in layer ℓ defines b_ℓ. We select

$$\mathcal{I}_\ell = \mathrm{TopK}_{j \in \mathcal{J}_\ell^{\mathrm{act}}}(-\bar{r}_j, b_\ell) \tag{4}$$

The operator TopK returns indices with the largest input values, so using $-\bar{r}_j$ selects the b_ℓ smallest active scores in layer ℓ. We set $M_p = 1$ if parameter p belongs to $\mathcal{G}_j$ for some selected $j \in \mathcal{I}_\ell$ in any layer ℓ, and set $M_p = 0$ otherwise. The server keeps M for fusion and sends it only to the forget client for masked local projection. Non-forget clients do not need this mask.

4.3 Projected Class Level Unlearning (PCLU)

PCLU coordinates target suppression with retained learning. Deletion and retained updates can oppose each other, so PCLU uses retained information to limit the size of deletion edits and reduce cancellation before fusion. PCLU follows the unlearning schedule

$$\mathcal{U} = (T_u, S, \omega_0)\,, \tag{5}$$

where T_u is the number of rounds, S is the number of local steps per round, and ω_0 is the initial forget weight. The schedule gives the forget weight ω_t at round t. The server broadcasts θ_{t-1} at round $t \in \{1, \ldots, T_u\}$. Each non-forget client $k \neq f$ initializes $\theta_k^{(t,0)} = \theta_{t-1}$, runs the local steps specified by $\mathcal{U}$ on $\mathcal{D}_k$, and uploads $\Delta_k^{(t)} = \theta_k^{(t,S)} - \theta_{t-1}$. Only the forget client uses the requested label c.

Forget client update construction. The forget client splits its local data into $\mathcal{D}_{\mathrm{del}} = \mathcal{D}_f^c$ and $\mathcal{D}_f^{\neg c} = \mathcal{D}_f \setminus \mathcal{D}_{\mathrm{del}}$, and initializes $\theta_f^{(t,0)} = \theta_{t-1}$. At local step $s \in \{0, \ldots, S-1\}$, it samples a deletion mini batch $B_{\mathrm{del}} \subset \mathcal{D}_{\mathrm{del}}$ and a retained mini batch $B_{\mathrm{ret}} \subset \mathcal{D}_f^{\neg c}$. Let $\theta = \theta_f^{(t,s)}$. The forget client uses a deletion objective $\mathcal{L}_{\mathrm{del}}$ on B_{del} and a retained objective $\mathcal{L}_{\mathrm{ret}}$ on B_{ret}. The deletion objective suppresses class c on deletion samples. The retained objective preserves non requested data with cross entropy, distillation from the broadcast model, and a squared distance anchor to the broadcast model. This reference model remains frozen throughout the local steps of the round. Appendix A gives the objective details. The gradients used by the local update are

$$g_{\mathrm{del}} = \nabla_\theta \mathcal{L}_{\mathrm{del}}(\theta, B_{\mathrm{del}}, c), \qquad g_{\mathrm{ret}} = \nabla_\theta \mathcal{L}_{\mathrm{ret}}(\theta, B_{\mathrm{ret}}, \theta_{t-1}) \tag{6}$$

For a multi class request $(f, \mathcal{C}_{\mathrm{del}})$, DeRe replaces $\mathcal{D}_f^c$ with the union of requested classes on client f. The deletion loss is averaged over these classes in the same unlearning round, while PBMS, PCLU, and MCR remain unchanged. Details are given in Appendix C.

The forget client applies projection only inside the selected mask. Set $\bar{g}_{\mathrm{del}} = g_{\mathrm{del}} \odot M$ and $\bar{g}_{\mathrm{ret}} = g_{\mathrm{ret}} \odot M$. If $\langle \bar{g}_{\mathrm{del}}, \bar{g}_{\mathrm{ret}} \rangle < 0$, the deletion direction is changed by

$$\bar{g}_{\mathrm{del}} \leftarrow \bar{g}_{\mathrm{del}} - \frac{\langle \bar{g}_{\mathrm{del}}, \bar{g}_{\mathrm{ret}} \rangle}{\|\bar{g}_{\mathrm{ret}}\|_2^2 + \epsilon} \bar{g}_{\mathrm{ret}} \tag{7}$$

This reduces opposition between the two masked gradients. If the inner product is non negative, $\bar{g}_{\mathrm{del}}$ is unchanged. The local direction g_{loc} and update step are

$$g_{\mathrm{loc}} = \omega_t \bar{g}_{\mathrm{del}} + g_{\mathrm{ret}}, \qquad \theta_f^{(t,s+1)} = \theta - \eta_{\mathrm{loc}} g_{\mathrm{loc}} \tag{8}$$

Here, η_{loc} is the local learning rate. The same ω_t is used throughout round t in local training and server fusion. The retained gradient acts on all parameters, while the deletion gradient is restricted to M. After S local steps, the forget client uploads $\Delta_f^{(t)} = \theta_f^{(t,S)} - \theta_{t-1}$, which contains both deletion and retained learning effects.

Server update fusion. The server receives $\Delta_f^{(t)}$ and $\{\Delta_k^{(t)} : k \neq f\}$. Let $\bar{\Delta}_{\neg f}^{(t)}$ be the mean non-forget update. The update protection rule $\mathcal{P}_M$ scales and clips the forget client update inside M, then reduces its overlap with the non-forget direction. It produces

$$\hat{\Delta}_f^{(t)} = \mathcal{P}_M\left(\Delta_f^{(t)}, \bar{\Delta}_{\neg f}^{(t)}\right). \tag{9}$$

The server rule $\mathcal{A}_M$ applies the mean non-forget update inside and outside the selected region. Appendix B gives both rules. The merged update is

$$\begin{aligned} \Delta_{\text{merge}}^{(t)} &= \omega_t \hat{\Delta}_f^{(t)} + \mathcal{A}_M\left(\bar{\Delta}_{\neg f}^{(t)}\right) \\ \theta_t &= \theta_{t-1} + \Delta_{\text{merge}}^{(t)} \end{aligned} \tag{10}$$

The server uses the same forget weight ω_t as the local update.

4.4 Mask Constrained Recovery (MCR)

MCR restores retained utility while protecting the selected edits. It starts from $\theta_0^{\text{rec}} = \theta_{T_u}$ and runs T_{rec} recovery rounds, with selected parameters frozen at the server to prevent their direct overwriting. At round $m \in \{1, \ldots, T_{\text{rec}}\}$, non-forget clients train on $\mathcal{D}_k$, while the forget client trains only on $\mathcal{D}_f^{\neg c}$. The forget weight is zero during recovery, so no deletion loss contributes to these updates. Each client uploads its model difference from the broadcast model $\theta_{m-1}^{\text{rec}}$. Let $\Delta_{\text{rec}}^{(m)}$ denote the server recovery update formed from the mean of these K client updates. The server applies only its outside mask part,

$$\theta_m^{\text{rec}} = \theta_{m-1}^{\text{rec}} + \Delta_{\text{rec}}^{(m)} \odot (1 - M) \tag{11}$$

After recovery, the final model is $\theta_{\text{post}} = \theta_{T_{\text{rec}}}^{\text{rec}}$.

5 Experiments

5.1 Experimental Setup

We evaluate on Digit5, DomainNet, and PACS. Following prior studies (Nguyen et al., 2026; Liao et al., 2025), each domain is treated as one client. Digit5 has five clients from MNIST (LeCun et al., 1998), SVHN (Netzer et al., 2011), USPS (Hull, 1994), SynthDigits (Ganin & Lempitsky, 2015), and MNIST-M (Ganin & Lempitsky, 2015). DomainNet has six clients from Clipart, Infograph, Painting, Quickdraw, Real, and Sketch. Motivated by the ten class DomainNet setting in FedBN (Li et al., 2021), we use a ten class DomainNet subset. PACS has four clients from art painting, cartoon, photo, and sketch. Client 0 is MNIST on Digit5, Clipart on DomainNet, and art painting on PACS. The main request is $(f = 0, c = 0)$, where $c = 0$ follows the processed label order of each dataset. This means removing digit 0 from MNIST on Digit5, bird from Clipart on DomainNet, and dog from art painting on PACS. Every non-forget client contains class c, so only $\mathcal{D}_f^c$ is requested for removal. We run three seeds, 0, 42, and 2048, and report the mean value. Detailed class lists, data splits, model choices, training settings, and the DeRe configuration are given in Appendix D and Appendix E.

We compare DeRe with Retrain, FCU (Deng et al., 2024), FedAU (Gu et al., 2024b), FedCDP (Wang et al., 2022), FedOSD (Pan et al., 2025), FOUL (Nguyen et al., 2026), and FUSED (Zhong et al., 2025). We use their released code and run all methods under the same client splits, request, and pretrained global model θ_{pre} in each run. They also follow the same privacy rule, where only the server and the forget client know the requested label c, and non-forget clients are not told c and continue ordinary local training. Retrain reruns federated training after removing $\mathcal{D}_f^c$ from client f. We report accuracy, running time, and membership inference attack success rate.

5.2 Evaluation Metrics

Following prior FU evaluations (Nguyen et al., 2026; Gu et al., 2024a; Khalil et al., 2025), we report FA and RA. Let $\mathcal{T}_k$ be the held out test set of client k, and let $\mathcal{T}_k^a = \{(x, y) \in \mathcal{T}_k \mid y = a\}$ be its samples from class a. For the main single class request, we use $\mathcal{T}_{\text{del}} = \mathcal{T}_f^c$ and $\mathcal{T}_{\text{ret}} = (\bigcup_{k=0}^{K-1} \mathcal{T}_k) \setminus \mathcal{T}_{\text{del}}$. FA and RA are the classification accuracy on $\mathcal{T}_{\text{del}}$ and $\mathcal{T}_{\text{ret}}$. The retained test set

Table 1: Single class unlearning results for request $(f = 0, c = 0)$. Retrain is the data deletion reference. FA_c is computed on $\mathcal{T}_f^c$, and $\Delta_{\mathrm{RF}} = \mathrm{RA} - \mathrm{FA}_c$. The FL row is θ_{pre}, and other rows are θ_{post}. Blue values show post minus FL changes, and bold marks the largest Δ_{RF}.

Single class unlearning request $(f = 0, c = 0)$, FA_c on $\mathcal{T}_f^c$														
Methods	Digit5					DomainNet					PACS				
	FA_c (%, ↓)		RA (%, ↑)		Δ_{RF} (%, ↑)	FA_c (%, ↓)		RA (%, ↑)		Δ_{RF} (%, ↑)	FA_c (%, ↓)		RA (%, ↑)		Δ_{RF} (%, ↑)
FL model	98.78		92.03			64.18		63.10			55.26		72.89		
Retrain	97.86	(-0.92)	91.45	(-0.58)	-6.41	41.79	(-22.39)	61.70	(-1.40)	19.91	25.00	(-30.26)	70.19	(-2.71)	45.19
FCU	97.76	(-1.02)	91.42	(-0.61)	-6.34	50.75	(-13.43)	61.05	(-2.05)	10.30	38.16	(-17.10)	72.58	(-0.31)	34.42
FedAU	0.00	(-98.78)	86.30	(-5.73)	86.30	0.00	(-64.18)	60.54	(-2.56)	60.54	0.00	(-55.26)	63.32	(-9.57)	63.32
FedCDP	88.27	(-10.51)	87.50	(-4.53)	-0.77	8.96	(-55.22)	59.08	(-4.02)	50.12	7.89	(-47.37)	66.81	(-6.08)	58.92
FedOSD	0.00	(-98.78)	8.17	(-83.86)	8.17	46.27	(-17.91)	33.21	(-29.89)	-13.06	30.26	(-25.00)	65.35	(-7.54)	35.09
FOUL	98.36	(-0.42)	91.30	(-0.73)	-7.06	61.19	(-2.99)	61.32	(-1.78)	0.13	21.05	(-34.21)	33.30	(-39.59)	12.25
FUSED	98.16	(-0.62)	91.19	(-0.84)	-6.97	61.63	(-2.55)	61.14	(-1.96)	-0.49	53.95	(-1.31)	72.46	(-0.43)	18.51
DeRe	0.00	(-98.78)	86.95	(-5.08)	**86.95**	0.00	(-64.18)	61.92	(-1.18)	**61.92**	0.00	(-55.26)	68.47	(-4.42)	**68.47**

is fixed before and after unlearning, so requested forget samples are excluded from RA while class c samples from non-forget clients remain in RA. Lower FA after unlearning means stronger target suppression, while higher RA means better overall retained utility.

All methods start from the same pretrained model θ_{pre}, and θ_{post} denotes the model after unlearning. We also report $\mathrm{FG} = \mathrm{FA}(\theta_{\mathrm{pre}}) - \mathrm{FA}(\theta_{\mathrm{post}})$, $\mathrm{GD} = \max\{0, \mathrm{RA}(\theta_{\mathrm{pre}}) - \mathrm{RA}(\theta_{\mathrm{post}})\}$, and $\Delta_{\mathrm{RF}} = \mathrm{RA}(\theta_{\mathrm{post}}) - \mathrm{FA}(\theta_{\mathrm{post}})$. Higher FG means a larger accuracy reduction on the requested forget set, and lower GD means a smaller accuracy loss on retained test data. The difference Δ_{RF} summarizes target suppression and overall retained accuracy for the stated request, with FA and RA reported separately to show the tradeoff. We also report MIA ASR (Shokri et al., 2017) and running time (Wang et al., 2025c). MIA details are given in Appendix F, and running time is measured from the start of a request to the output of θ_{post}, including all stages of each method.

5.3 Main Results

Unlearning performance. Table 1 compares target suppression and overall retained accuracy. Retrain provides the data deletion reference, while equal FA permits comparing retained utility.

Retrain keeps RA close to the FL model. Its training set $\mathcal{D}_{\mathrm{ret}}$ excludes the requested samples but retains class c on non-forget clients, allowing target recognition through retained data. DeRe adds suppression on the forget client's domain and reaches zero FA. Compared with Retrain, its RA is lower on Digit5 and PACS and slightly higher on DomainNet, reflecting the added suppression goal.

The other methods show why suppression and retained utility must be assessed together. FCU and FUSED maintain high RA with substantial target recognition, as does FOUL on Digit5 and DomainNet. At the other extreme, FedOSD reaches zero FA on Digit5 with a large RA loss. These outcomes expose the tension between suppressing behavior on the forget client's domain and preserving knowledge used by retained data, motivating control of the edit region and its updates.

FedAU and DeRe both reach $\mathrm{FA}_c = 0.00$ on all three datasets. At this matched target suppression, DeRe retains higher overall RA and obtains the largest Δ_{RF} among the compared methods. This is consistent with the design of DeRe, which limits deletion to the selected region, coordinates updates before fusion, and recovers utility outside the mask. Ablations examine these contributions.

Figure 4 expresses the same comparison through FG and GD. Low GD alone can accompany little target suppression, so preservation must be judged alongside forgetting. Across the three datasets, DeRe combines the largest observed FG with lower GD than the other methods reaching that FG, supporting its ability to limit retained accuracy loss at matched suppression.

Multi class request. We further test removing classes 0, 1, and 2 from client 0. This means removing digits 0, 1, and 2 from MNIST on Digit5, bird, feather, and headphones from Clipart on DomainNet, and dog, elephant, and giraffe from art painting on PACS. Other clients retain all samples from these classes. The request seeks to suppress their recognition on client 0 while preserving recognition in other domains and accuracy on the remaining classes.

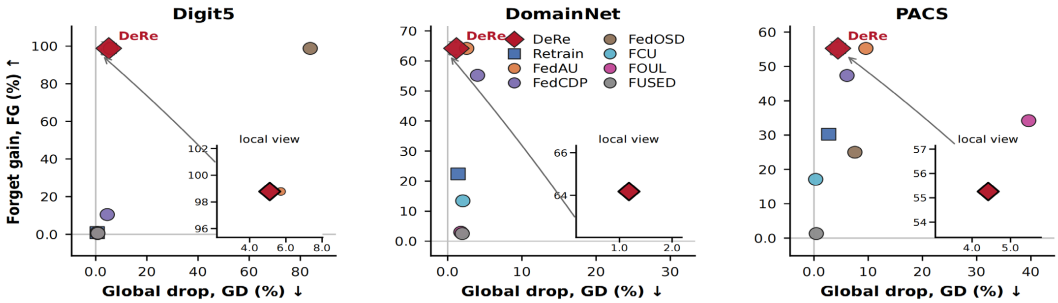

Figure 4: Forgetting and retained accuracy change. Each point is one method. The upper left region is better because it means larger FG and smaller GD.

Table 2: Multi class unlearning results for request $(f = 0, \mathcal{C}_{\text{del}} = \{0, 1, 2\})$. Retrain is the data deletion reference. $\text{FA}_{\mathcal{C}}$ is computed on $\bigcup_{c \in \mathcal{C}_{\text{del}}} \mathcal{T}_f^c$, and $\Delta_{\text{RF}} = \text{RA} - \text{FA}_{\mathcal{C}}$. The FL model row is θ_{pre}, and other rows are θ_{post}. Blue values show post minus FL model changes for $\text{FA}_{\mathcal{C}}$ and RA.

Multi class unlearning request $(f = 0, \mathcal{C}_{\text{del}} = \{0, 1, 2\})$, $\text{FA}_{\mathcal{C}}$ on $\bigcup_{c \in \mathcal{C}_{\text{del}}} \mathcal{T}_f^c$														
Methods	Digit5					DomainNet					PACS				
	$\text{FA}_{\mathcal{C}}$ (%, ↓)		RA (%, ↑)		Δ_{RF} (%, ↑)	$\text{FA}_{\mathcal{C}}$ (%, ↓)		RA (%, ↑)		Δ_{RF} (%, ↑)	$\text{FA}_{\mathcal{C}}$ (%, ↓)		RA (%, ↑)		Δ_{RF} (%, ↑)
FL model	99.05		91.70			62.92		63.12			57.61		73.70		
Retrain	98.35	(-0.70)	90.65	(-1.05)	-7.70	56.74	(-6.18)	61.09	(-2.03)	4.35	28.80	(-28.81)	65.38	(-8.32)	36.58
FCU	95.65	(-3.40)	90.96	(-0.74)	-4.69	52.25	(-10.67)	57.38	(-5.74)	5.13	48.76	(-8.85)	72.38	(-1.32)	23.62
FedAU	20.91	(-78.14)	67.43	(-24.27)	46.52	5.06	(-57.86)	53.39	(-9.73)	48.33	4.35	(-53.26)	60.92	(-12.78)	56.57
FedCDP	96.82	(-2.23)	90.14	(-1.56)	-6.68	35.96	(-26.96)	59.39	(-3.73)	23.43	8.15	(-49.46)	64.61	(-9.09)	56.46
FedOSD	0.00	(-99.05)	8.54	(-83.16)	8.54	33.71	(-29.21)	35.03	(-28.09)	1.32	35.87	(-21.74)	69.51	(-4.19)	33.64
FOUL	99.27	(+0.22)	90.02	(-1.68)	-9.25	63.48	(+0.56)	61.28	(-1.84)	-2.20	30.43	(-27.18)	33.08	(-40.62)	2.65
FUSED	98.79	(-0.26)	89.26	(-2.44)	-9.53	61.42	(-1.50)	62.03	(-1.09)	0.61	55.61	(-2.00)	71.37	(-2.33)	15.76
DeRe	17.27	(-81.78)	90.79	(-0.91)	**73.52**	3.03	(-59.89)	57.86	(-5.26)	**54.83**	7.33	(-50.28)	67.48	(-6.22)	**60.15**

Table 2 tests the same procedure with several classes requested together. Retrain still recognizes these classes on the forget client's domain after data deletion, consistent with their continued presence on other clients. DeRe lowers $\text{FA}_{\mathcal{C}}$ on all three datasets, with lower RA than Retrain on DomainNet, similar RA on Digit5, and higher RA on PACS. The need to coordinate target suppression with retained learning therefore remains when the request covers multiple classes.

Compared with FedAU, DeRe has lower target accuracy and higher RA on Digit5 and DomainNet. On PACS, FedAU suppresses more strongly but retains lower RA. DeRe obtains the largest Δ_{RF} across the three datasets, although target recognition remains above zero.

The requested classes jointly determine the probe responses and deletion direction, while retained updates guide the scaling and projection of the resulting forget client update. This lets update coordination account for the combined deletion effect as other clients continue learning those same classes. Suppression and RA are consistent with using retained information to control the utility loss of a broader deletion request.

Unlearning cost. Table 3 reports the time to forget. DeRe is not the fastest method because it includes probe, PCLU, and recovery, while FedAU is faster but has larger retained accuracy loss at the same forgetting level in Table 1 and Figure 4. Compared with Retrain, which reruns federated training on $\mathcal{D}_{\text{ret}}$, DeRe is about $2.4\times$, $5.5\times$, and $10.5\times$ faster on Digit5, DomainNet, and PACS.

Table 3: Time to forget in seconds, excluding evaluation.

Methods	**Digit5**	**DomainNet**	**PACS**
Retrain	644.77	1512.58	817.94
FCU	71.90	52.55	75.98
FedAU	6.57	8.09	6.21
FedCDP	138.35	60.19	37.80
FedOSD	416.73	838.55	445.68
FOUL	331.73	196.79	111.30
FUSED	697.90	451.53	337.49
DeRe	264.54	272.73	78.08

Table 4: MIA ASR after unlearning. Random guessing corresponds to 0.5.

Methods	**Digit5**	**DomainNet**	**PACS**
Retrain	0.490	0.474	0.491
FCU	0.492	0.457	0.684
FedAU	0.504	0.637	0.680
FedCDP	0.524	0.521	0.571
FedOSD	0.487	0.558	0.575
FOUL	0.514	0.593	0.440
FUSED	0.492	0.605	0.638
DeRe	0.486	0.517	0.564

Table 5: Ablation with cumulative stage additions, PCLU removal, and a random mask using the same per layer budgets as PBMS. Time excludes evaluation.

Variants	Digit5			DomainNet			PACS		
	FG (%, ↑)	GD (%, ↓)	Time (s)	FG (%, ↑)	GD (%, ↓)	Time (s)	FG (%, ↑)	GD (%, ↓)	Time (s)
Base	98.78	5.77	164.06	64.18	52.21	124.29	55.26	45.61	44.56
+ PBMS	98.78	5.71	198.56	64.18	44.36	180.04	55.26	41.00	53.65
+ PBMS+PCLU	98.78	5.15	226.13	64.18	1.58	193.83	55.26	5.41	55.35
+ PBMS+PCLU+MCR (DeRe)	98.78	5.08	264.54	64.18	1.18	272.73	55.26	4.42	78.08
+ PBMS+MCR	98.78	5.94	237.87	63.29	7.07	258.60	51.63	11.06	76.98
Random mask+PCLU+MCR	35.71	1.46	218.07	34.33	0.47	218.99	46.05	1.30	72.44

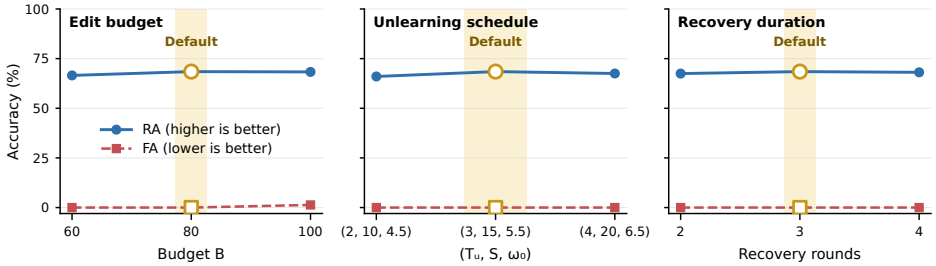

Figure 5: Parameter sensitivity of DeRe on PACS. Yellow bands mark the default settings.

MIA ASR. Table 4 reports post-unlearning MIA ASR. MIA ASR and FA measure different behavior. Retrain shows this difference because it has low ASR but can still recognize $\mathcal{T}_{\text{del}}$ using class c from non-forget clients. For DeRe, ASR is close to random guessing on Digit5 and DomainNet, and is lower than FedAU on all datasets while reaching the same FA. This suggests that the lower GD of DeRe does not increase this attack success.

5.4 Ablation Study

Component ablation. Table 5 evaluates cumulative additions and targeted removals. Replacing PBMS with a random mask weakens target suppression, confirming that response based selection is needed to locate deletion specific groups; its lower GD reflects weak forgetting rather than a better tradeoff. PCLU provides the main utility gain by coordinating deletion and retained updates before fusion, while PBMS+MCR shows that recovery cannot replace this coordination. MCR then reduces the remaining utility loss by updating only outside M without overwriting the selected edits. Thus, PBMS locates the edit, PCLU controls update interaction, and MCR restores utility while preserving it; all three are necessary to the full method.

Parameter sensitivity. Figure 5 shows stable target suppression and retained utility as $B, \mathcal{U} = (T_u, S, \omega_0)$, and T_{rec} vary. PBMS preserves deletion specific ranking, PCLU coordinates deletion and retained updates across schedules, and MCR keeps recovery outside M. This separation makes DeRe robust to its key configurations rather than dependent on narrowly tuned settings.

6 Conclusion

We studied FU under cross-domain feature shift to suppress target behavior in one domain while preserving related capabilities elsewhere. Other clients' class knowledge may combine with the forget client's retained domain information to support that behavior. DeRe selects an edit region, coordinates deletion and retained updates at the client and server, and recovers utility outside the mask without directly overwriting selected edits. Experiments on Digit5, DomainNet, and PACS show a better balance between suppression and overall retained accuracy for single and multi class requests under the evaluated protocol. Ablations support probe selection for suppression, update coordination for limiting retained accuracy loss, and recovery for restoring utility.

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

Jun Bai, Yiliao Song, Di Wu, Atul Sajjanhar, Yong Xiang, Wei Zhou, Xiaohui Tao, Yan Li, and Yue Li. A unified solution to diverse heterogeneities in one-shot federated learning. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 71–82. ACM, 2025. doi: 10.1145/3711896.3736825.

Tianshi Che, Yang Zhou, Zijie Zhang, Lingjuan Lyu, Ji Liu, Da Yan, Dejing Dou, and Jun Huan. Fast federated machine unlearning with nonlinear functional theory. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 4241–4268. PMLR, 2023. URL https://proceedings.mlr.press/v202/che23b.html.

Minghui Chen, Meirui Jiang, Xin Zhang, Qi Dou, Zehua Wang, and Xiaoxiao Li. Local superior soups: A catalyst for model merging in cross-silo federated learning. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-0657.

Vikram S. Chundawat, Pushkar Niroula, Prasanna Dhungana, Stefan Schoepf, Murari Mandal, and Alexandra Brintrup. ConDa: Fast federated unlearning with contribution dampening. arXiv preprint arXiv:2410.04144, 2024.

Zhipeng Deng, Luyang Luo, and Hao 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. Springer Nature Switzerland, 2024. doi: 10.1007/978-3-031-72117-5_23.

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.

Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1180–1189, Lille, France, 2015. PMLR. URL https://proceedings.mlr.press/v37/ganin15.html.

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

Hanlin Gu, Win Kent Ong, Chee Seng Chan, and Lixin Fan. Ferrari: Federated feature unlearning via optimizing feature sensitivity. *Advances in Neural Information Processing Systems*, 37:24150–24180, 2024a.

Hanlin Gu, Gongxi Zhu, Jie Zhang, Xinyuan Zhao, Yuxing Han, Lixin Fan, and Qiang Yang. Unlearning during learning: An efficient federated machine unlearning method. In Kate Larson (ed.), *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pp. 4035–4043. International Joint Conferences on Artificial Intelligence Organization, August 2024b. doi: 10.24963/ijcai.2024/446. Main Track.

Hao Guan, Pew-Thian Yap, Andrea Bozoki, and Mingxia Liu. Federated learning for medical image analysis: A survey. *Pattern Recognition*, 151:110424, 2024. doi: 10.1016/j.patcog.2024.110424.

Jonathan 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.

Daniel M. Jimenez G., David Solans, Mikko A. Heikkilä, Andrea Vitaletti, Nicolas Kourtellis, Aris Anagnostopoulos, and Ioannis Chatzigiannakis. Non-IID data in federated learning: A survey with taxonomy, metrics, methods, frameworks and future directions. arXiv preprint arXiv:2411.12377, 2024.

Daniel M. Jimenez-Gutierrez, Mehrdad Hassanzadeh, Aris Anagnostopoulos, Ioannis Chatzigiannakis, and Andrea Vitaletti. A thorough assessment of the Non-IID data impact in federated learning. *Journal of Industrial Information Integration*, 50:101052, 2026. doi: 10.1016/j.jii.2025.101052.

Yasser H. Khalil, Leo Brunswic, Soufiane Lamghari, Xu Li, Mahdi Beitollahi, and Xi Chen. NoT: Federated unlearning via weight negation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 25759–25769, June 2025. doi: 10.1109/CVPR52734.2025.02399.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwińska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences of the United States of America*, 114(13):3521–3526, 2017. doi: 10.1073/pnas.1611835114.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 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. 5542–5550, 2017. doi: 10.1109/ICCV.2017.591.

Guanghao Li, Li Shen, Yan Sun, Yue Hu, Han Hu, and Dacheng Tao. Subspace based federated unlearning. *Transactions on Machine Learning Research*, 2025a. URL https://openreview.net/forum?id=KE2ZNl2lFP.

Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou. FedBN: Federated learning on Non-IID features via local batch normalization. In *9th International Conference on Learning Representations*. OpenReview.net, 2021. URL https://openreview.net/forum?id=6YEQUn0QICG.

Yuyuan Li, Jiaming Zhang, Yixiu Liu, and Chaochao Chen. Class-wise federated unlearning: Harnessing active forgetting with teacher–student memory generation. *Knowledge-Based Systems*, 316:113353, 2025b. doi: 10.1016/j.knosys.2025.113353.

Tianchi Liao, Binghui Xie, Lele Fu, Sheng Huang, Bowen Deng, Chuan Chen, and Zibin Zheng. Federated domain generalization with decision insight matrix. In James Kwok (ed.), *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 5689–5697. International Joint Conferences on Artificial Intelligence Organization, August 2025. doi: 10.24963/ijcai.2025/633. Main Track.

Yijing Lin, Zhipeng Gao, Hongyang Du, Dusit Niyato, Gui Gui, Shuguang Cui, and Jinke Ren. Scalable federated unlearning via isolated and coded sharding. In Kate Larson (ed.), *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pp. 4551–4559. International Joint Conferences on Artificial Intelligence Organization, 2024. doi: 10.24963/ijcai.2024/503.

Gaoyang Liu, Xiaoqiang Ma, Yang Yang, Chen Wang, and Jiangchuan Liu. FedEraser: Enabling efficient client-level data removal from federated learning models. In *Proceedings of the 2021 IEEE/ACM 29th International Symposium on Quality of Service*, pp. 1–10. IEEE, 2021. doi: 10.1109/IWQOS52092.2021.9521274.

Yi Liu, Lei Xu, Xingliang Yuan, Cong Wang, and Bo Li. The right to be forgotten in federated learning: An efficient realization with rapid retraining. In *Proceedings of IEEE INFOCOM 2022 – IEEE Conference on Computer Communications*, pp. 1749–1758. IEEE, 2022. doi: 10.1109/INFOCOM48880.2022.9796721.

Ziyao Liu, Yu Jiang, Jiyuan Shen, Minyi Peng, Kwok-Yan Lam, Xingliang Yuan, and Xiaoning Liu. A survey on federated unlearning: Challenges, methods, and future directions. *ACM Computing Surveys*, 57(1):1–38, 2024. doi: 10.1145/3679014.

Jakub Łucki, Boyi Wei, Yangsibo Huang, Peter Henderson, Florian Tramèr, and Javier Rando. An adversarial perspective on machine unlearning for AI safety. *Transactions on Machine Learning Research*, 2025. URL https://openreview.net/forum?id=J5IRyTKZ9s.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *Proceedings of the NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011. URL https://ai.stanford.edu/~twangcat/papers/nips2011_housenumbers.pdf.

A. Tuan Nguyen, Philip Torr, and Ser-Nam Lim. FedSR: A simple and effective domain generalization method for federated learning. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://openreview.net/forum?id=mrt90D00aQX.

Minh-Duong Nguyen, Senura Wanasekara, Le-Tuan Nguyen, Quoc-Viet Pham, Ken-Tye Yong, Nguyen H. Tran, and Dung D. Le. Computation and communication efficient federated unlearning via on-server gradient conflict mitigation and expression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3347–3357, 2026.

Win Kent Ong and Chee Seng Chan. Maverick: Collaboration-free federated unlearning for medical privacy. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2025*, volume 15973, pp. 358–368. Springer Nature Switzerland, 2025.

Zibin Pan, Zhichao Wang, Chi Li, Kaiyan Zheng, Boqi Wang, Xiaoying Tang, and Junhua 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.v39i19.34181.

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, October 2019. doi: 10.1109/ICCV.2019.00149.

Yixuan Qiu, Shaofei Shen, Chenhao Zhang, Lin Yue, Weitong Chen, and Miao Xu. Margin-aware prototype learning for client withdrawal in federated unlearning. *Neural Networks*, 204:109231, 2026. doi: 10.1016/j.neunet.2026.109231.

Nicolò Romandini, Alessio Mora, Carlo Mazzocca, Rebecca Montanari, and Paolo Bellavista. Federated unlearning: A survey on methods, design guidelines, and evaluation metrics. *IEEE Transactions on Neural Networks and Learning Systems*, 36(7):11697–11717, 2025. doi: 10.1109/TNNLS.2024.3478334.

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

Youming Tao, Cheng-Long Wang, Miao Pan, Dongxiao Yu, Xiuzhen Cheng, and Di Wang. Communication efficient and provable federated unlearning. *Proceedings of the VLDB Endowment*, 17(5):1119–1131, 2024. doi: 10.14778/3641204.3641220.

Tim van Erven and Peter Harremoës. Rényi divergence and Kullback-Leibler divergence. *IEEE Transactions on Information Theory*, 60(7):3797–3820, 2014. doi: 10.1109/TIT.2014.2320500.

Changsheng Wang, Yihua Zhang, Jinghan Jia, Parikshit Ram, Dennis Wei, Yuguang Yao, Soumyadeep Pal, Nathalie Baracaldo, and Sijia Liu. Invariance makes LLM unlearning resilient even to unanticipated downstream fine-tuning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 65464–65479. PMLR, 2025a. URL https://proceedings.mlr.press/v267/wang25en.html.

Junxiao Wang, Song Guo, Xin Xie, and Heng 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.

Meng Wang, Kai Yu, Chun-Mei Feng, Yiming Qian, Ke Zou, Lianyu Wang, Rick Siow Mong Goh, Xinxing Xu, Yong Liu, and Huazhu Fu. Reliable federated disentangling network for non-IID domain feature. *IEEE Transactions on Big Data*, 11(2):648–658, 2025b. doi: 10.1109/TBDATA.2024.3423694.

Weiqi Wang, Chenhan Zhang, Zhiyi Tian, and Shui Yu. FedU: Federated unlearning via user-side influence approximation forgetting. *IEEE Transactions on Dependable and Secure Computing*, 22 (3):2550–2562, 2025c. doi: 10.1109/TDSC.2024.3520614.

Chen Wu, Sencun Zhu, and Prasenjit Mitra. Federated unlearning with knowledge distillation. arXiv preprint arXiv:2201.09441, 2022.

Di Wu, Jun Bai, Yiliao Song, Junjun Chen, Wei Zhou, Yong Xiang, and Atul Sajjanhar. FedInverse: Evaluating privacy leakage in federated learning. In *Proceedings of the Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=nTNgkEIfeb.

Wentai Wu, Hanwei Tan, Yijun Quan, Haixia Peng, Ligang He, Bin Yang, and C. L. Philip Chen. Lethe: Principled dual-stream update for persistent knowledge erasure in federated unlearning. arXiv preprint arXiv:2601.22601, 2026. URL https://arxiv.org/abs/2601.22601v3.

Xinwei Wu, Junzhuo Li, Minghui Xu, Weilong Dong, Shuangzhi Wu, Chao Bian, and Deyi 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, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.174.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 5824–5836, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/3fe78a8acf5fda99de95303940a2420c-Abstract.html.

Zhengyi Zhong, Weidong Bao, Ji Wang, Shuai Zhang, Jingxuan Zhou, Lingjuan Lyu, and Wei Yang Bryan Lim. Unlearning through knowledge overwriting: Reversible federated unlearning via selective sparse adapter. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 30661–30670, June 2025. doi: 10.1109/CVPR52734.2025.02855.

A PCLU Loss Details

The forget client uses deletion and retained objectives, with notation from Section 4. At local step s of unlearning round t, let $\theta = \theta_f^{(t,s)}$. The forget client samples $B_{\mathrm{del}} \subset \mathcal{D}_{\mathrm{del}}$ and $B_{\mathrm{ret}} \subset \mathcal{D}_f^{\neg c}$.

The deletion objective combines margin suppression, negative learning, negative entropy, and KL redistribution (van Erven & Harremoës, 2014) over the non requested classes. Let ℓ_{mbs}, ℓ_{nl}, ℓ_{ent}, and ℓ_{red} denote the scaled batch terms used in optimization. The retained objective combines cross entropy, distillation from the frozen broadcast model, and a squared distance anchor to that model. Let ℓ_{ce}, ℓ_{kd}, and ℓ_{anc} denote the corresponding scaled batch terms. The objectives are

$$\begin{aligned} \mathcal{L}_{\mathrm{del}} &= \ell_{\mathrm{mbs}} + \ell_{\mathrm{nl}} + \ell_{\mathrm{ent}} + \ell_{\mathrm{red}} \\ \mathcal{L}_{\mathrm{ret}} &= \ell_{\mathrm{ce}} + \ell_{\mathrm{kd}} + \ell_{\mathrm{anc}} \end{aligned} \tag{12}$$

These objectives produce g_{del} and g_{ret} in Section 4.

B Update Rule Details

Probe scoring. PBMS excludes a group when its deletion and non-forget responses are numerically zero in every probe run. The remaining groups form $\mathcal{J}_\ell^{\mathrm{act}}$ in layer ℓ.

PCLU update. The server uses retained updates to adjust both the size and direction of the forget client update. It computes the mean non-forget update $\bar{\Delta}_{\neg f}^{(t)}$ and the squared update statistic $F_p^{(t)}$ for parameter p,

$$\begin{aligned} \bar{\Delta}_{\neg f}^{(t)} &= \frac{1}{K-1} \sum_{k \neq f} \Delta_k^{(t)} \\ F_p^{(t)} &= \mathrm{EMA}_{r=1}^{t} \left[\frac{1}{K-1} \sum_{k \neq f} \left(\Delta_k^{(r)}(p) \right)^2 \right], \qquad F_p^{(0)} = 0 \end{aligned} \tag{13}$$

Here, r indexes unlearning rounds, and $\Delta_k^{(r)}(p)$ is the update at parameter p in round r. The operator EMA denotes an exponential moving average over these rounds. The statistic $F_p^{(t)}$ tracks ordinary update size across rounds and is computed from the received updates. The rule $\mathcal{P}_M$ uses this statistic to scale the forget client update inside M. It clips the masked norm using the median non-forget norm and reduces overlap with the non-forget direction. The processed update is

$$\hat{\Delta}_f^{(t)} = \mathcal{P}_M \left(\Delta_f^{(t)}, \bar{\Delta}_{\neg f}^{(t)} \right). \tag{14}$$

The server rule $\mathcal{A}_M$ keeps the mean non-forget update inside M and scales its part outside M.

C Multi Class Unlearning Details

We describe how DeRe handles a request that removes several classes from one forget client. Let the request be $(f, \mathcal{C}_{\mathrm{del}})$, where $\mathcal{C}_{\mathrm{del}} \subseteq \mathcal{Y}$ is the set of requested classes. The deletion set and the forget client's retained set are

$$\mathcal{D}_{\mathrm{del}}^{\mathcal{C}} = \bigcup_{b \in \mathcal{C}_{\mathrm{del}}} \mathcal{D}_f^b, \qquad \mathcal{D}_f^{\neg \mathcal{C}} = \mathcal{D}_f \setminus \mathcal{D}_{\mathrm{del}}^{\mathcal{C}}. \tag{15}$$

Here, b denotes a class index in the requested class set. All samples from $\mathcal{D}_{\mathrm{del}}^{\mathcal{C}}$ are used as deletion samples. The forget client uses $\mathcal{D}_f^{\neg \mathcal{C}}$ as its retained data. Other clients keep all their local data, including samples from classes in $\mathcal{C}_{\mathrm{del}}$.

The multi class request is optimized in one unlearning procedure. It is not run as several single class requests. For a deletion mini batch $B_{\mathrm{del}} \subset \mathcal{D}_{\mathrm{del}}^{\mathcal{C}}$, let

$$B_{\mathrm{del}}^b = \{(x, y) \in B_{\mathrm{del}} \mid y = b\}, \qquad \mathcal{C}_B = \{b \in \mathcal{C}_{\mathrm{del}} \mid B_{\mathrm{del}}^b \neq \emptyset\}. \tag{16}$$

Table 6: FL training parameters for the pretrained model θ_{pre}. LR denotes local learning rate.

Dataset	Model	Client domains	Classes	Rounds	Batch size	LR	LR schedule
Digit5	ResNet18	MNIST, MNIST-M, SVHN, USPS, synthetic digits	10	200	128	0.05	cosine
DomainNet	ResNet18	Clipart, Infograph, Painting, Quickdraw, Real, Sketch	10	200	64	0.05	cosine
PACS	UResNet18	art painting, cartoon, photo, sketch	7	200	10	0.005	cosine

The set $\mathcal{C}_B$ contains the requested classes that appear in the mini batch. For each $b \in \mathcal{C}_B$, we compute the single class deletion loss in Appendix A with c replaced by b and B_{del} replaced by B^b_{del}. We denote this loss by $\mathcal{L}^{(b)}_{\mathrm{del}}$. The multi class deletion loss is

$$\mathcal{L}^{\mathcal{C}}_{\mathrm{del}} = \frac{1}{|\mathcal{C}_B|} \sum_{b \in \mathcal{C}_B} \mathcal{L}^{(b)}_{\mathrm{del}}. \tag{17}$$

This average makes each target class contribute to the deletion direction when it appears in the mini batch. The retained loss is unchanged, except that the retained mini batch is sampled from $\mathcal{D}_f^{\neg\mathcal{C}}$. Thus,

$$\begin{aligned} &g_{\mathrm{del}} = \nabla_\theta \mathcal{L}^{\mathcal{C}}_{\mathrm{del}}, \qquad g_{\mathrm{ret}} = \nabla_\theta \mathcal{L}_{\mathrm{ret}}(\theta, B_{\mathrm{ret}}, \theta_{t-1}), \\ &B_{\mathrm{ret}} \subset \mathcal{D}_f^{\neg\mathcal{C}}. \end{aligned} \tag{18}$$

PBMS uses the same rule as the single class setting. The only change is that the forget client probe is computed on $\mathcal{D}^{\mathcal{C}}_{\mathrm{del}}$. The selected mask therefore marks neurons that respond more to the requested class set than to non-forget client updates. PCLU uses the same update rule as the single class setting. MCR is also unchanged. The forget client recovers only on $\mathcal{D}_f^{\neg\mathcal{C}}$, and the server applies recovery updates only outside the selected mask.

For evaluation, the multi class forget test set is

$$\mathcal{T}^{\mathcal{C}}_{\mathrm{del}} = \bigcup_{b \in \mathcal{C}_{\mathrm{del}}} \mathcal{T}^b_f. \tag{19}$$

We report

$$\mathrm{FA}_{\mathcal{C}}(\theta) = \mathrm{Acc}(\theta; \mathcal{T}^{\mathcal{C}}_{\mathrm{del}}), \qquad \Delta_{\mathrm{RF}} = \mathrm{RA}(\theta_{\mathrm{post}}) - \mathrm{FA}_{\mathcal{C}}(\theta_{\mathrm{post}}). \tag{20}$$

RA is computed on the retained test set, which excludes $\mathcal{T}^{\mathcal{C}}_{\mathrm{del}}$ and keeps samples from the same classes on non-forget clients.

D Experimental Parameter Details

All unlearning methods start from the same FL model for each dataset. The settings below cover FL training and single class DeRe runs, and the multi class runs use the same FL models.

D.1 FL Training Parameters

Each domain is treated as one client. We use non-IID cross-domain data and do not split one domain into multiple clients. All clients participate in each FL round. The local optimizer is SGD with momentum 0.9 and weight decay 5×10^{-4}. Each client runs one local epoch per communication round. The FL models are trained from random initialization.

Digit5, DomainNet, and PACS use a cosine learning rate schedule. The cosine period is set to 10, and the minimum learning rate is 10^{-5}. Digit5 and DomainNet use dropout 0.1, while PACS uses dropout 0.0. The test split for each domain is kept as the client test set. The native training split is divided into local training and validation sets with an 80 to 20 ratio. All experiments are run on one NVIDIA RTX 4090 with 22GB memory.

Table 7: Single class DeRe configuration.

Setting	Digit5	DomainNet	PACS
Edit budget B	75	75	80
Unlearning schedule $\mathcal{U}$	$(3, 40, 1.50)$	$(2, 15, 3.50)$	$(3, 15, 5.50)$
Recovery duration T_{rec}	1	2	3

Table 8: Dataset statistics. Each domain is treated as one client.

Dataset	Clients	Domains	Classes	Train set	Test set	Total
Digit5	5	5	10	195,348	51,239	246,587
DomainNet	6	6	10	18,283	4,568	22,851
PACS	4	4	7	7,993	1,998	9,991

D.2 Single Class DeRe Configuration

The single class request uses forget client 0, forget class 0, and the delete strategy. Requested samples are used locally for deletion updates, while retained training excludes them. The unlearning schedule is $\mathcal{U} = (T_u, S, \omega_0)$, with $\omega_t = \omega_0(0.4)^{t-1}$ at round t.

E Dataset and Model Details

This section gives the dataset and model details used in Section 5.1. Digit5 uses five domains as clients, namely MNIST, SVHN, USPS, SynthDigits, and MNIST-M. Client 0 is MNIST. DomainNet uses six domains as clients, namely Clipart, Infograph, Painting, Quickdraw, Real, and Sketch. Client 0 is Clipart. The ten DomainNet classes are bird, feather, headphones, ice cream, teapot, tiger, whale, windmill, wine glass, and zebra. PACS uses art painting, cartoon, photo, and sketch as clients 0 to 3. Table 8 summarizes the dataset statistics.

For each client, the native test split of its domain is used as $\mathcal{T}_k$. The native training split is divided into local training and validation parts with 80 percent and 20 percent of samples. Digit5 and DomainNet use ResNet18, and PACS uses UResNet18 following the released FOUL code.

F Privacy Attack Details

We use a score based black box membership inference attack. The attacker queries the model output and computes the per sample cross entropy loss. It then applies a loss based threshold rule, where lower loss indicates a member prediction. The member set is the deleted training subset $\mathcal{D}_f^c$, and the non member set is the forget client test subset $\mathcal{T}_f^c$. Since the two sets may have different sizes, we report balanced ASR as

$$\text{ASR} = \frac{1}{2}\left(\text{Acc}_{\text{mem}} + \text{Acc}_{\text{non}}\right),$$

where Acc_{mem} and Acc_{non} are attack accuracies on member and non member samples.