ERPFU: PROACTIVE FEDERATED UNLEARNING WITH FIXED-SUPPORT CLASSIFIER ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated deletion requests require updating a trained model while preserving utility on retained data. We study how far request-time optimization can be restricted: can a fixed subset of the final classifier retain the utility of full retraining? Efficient and Reversible Proactive Federated Unlearning (ERPFU) combines training-time weight orthogonality regularization with a fixed-support sparse classifier adapter. After a request, clients optimize classifier residuals on retained data while keeping the encoder and base classifier frozen. This restriction makes the update budget explicit and admits a conditional analysis separating optimization error, excluded correction coordinates, and the gap between frozen-feature adaptation and full retraining. We also characterize the effects of input anisotropy and client averaging on weight-based preparation. Across four IID client-deletion dataset–backbone settings, ERPFU's retained accuracy differs from retraining by at most 1.85 percentage points. Shared-class and multi-client results further characterize utility preservation, while the CIFAR-10 gap grows to 4.66 points under a non-IID partition. These results support classifier-restricted adaptation as a utility-preserving option in the evaluated settings; the functional analysis does not certify privacy erasure.

1 INTRODUCTION

Federated unlearning (FU) updates a shared model after clients request removal of training data (Liu et al., 2021; 2022). Retraining from scratch on the retained data provides a natural reference, but repeats optimization of the complete model. In a federated system, this also involves coordination with retained clients. A practical question is how much of the model must remain trainable after a request to preserve the utility of this reference.

Restricting the update space creates a specific trade-off. A small trainable component reduces the optimization state, but it can express only corrections supported by the frozen model. Removing a client's data can also remove most examples of a class while leaving that class present elsewhere. An update must then change predictions associated with the request without unnecessarily degrading the shared and retained classes. The relevant question is therefore whether the restricted predictor preserves useful behavior, and under what conditions its predictions can approach those of retraining.

Existing FU methods address request-time adaptation through update reconstruction, parameter correction, or sparse adapters (Liu et al., 2021; Wu et al., 2022; Pan et al., 2025; Zhong et al., 2025). Proactive approaches additionally modify the initial training stage: PFU-SKLA combines orthogonal preparation with sensitivity-guided sparse adaptation across selected layers (Lai et al., 2026), while FOUL prepares the model for server-side unlearning (Nguyen et al., 2026). Figure 1 contrasts two choices of adaptation support. Both freeze the base weights, but encoder-layer adapters can change the feature transformation, whereas classifier-only adaptation fixes that transformation. This raises our central question: can a fixed subset of classifier coordinates preserve the utility of retraining? In ERPFU, the selected coordinates form a shared, fixed support throughout each request. This makes the trainable dimension explicit while leaving a capacity constraint: a fixed encoder may not support every correction that full retraining can produce.

Our starting point is that this restriction makes the approximation problem explicit. With frozen features, retained-data cross-entropy and quadratic residual regularization define a strongly convex classifier objective. Its solution can still differ from full retraining for two distinct reasons: the

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.

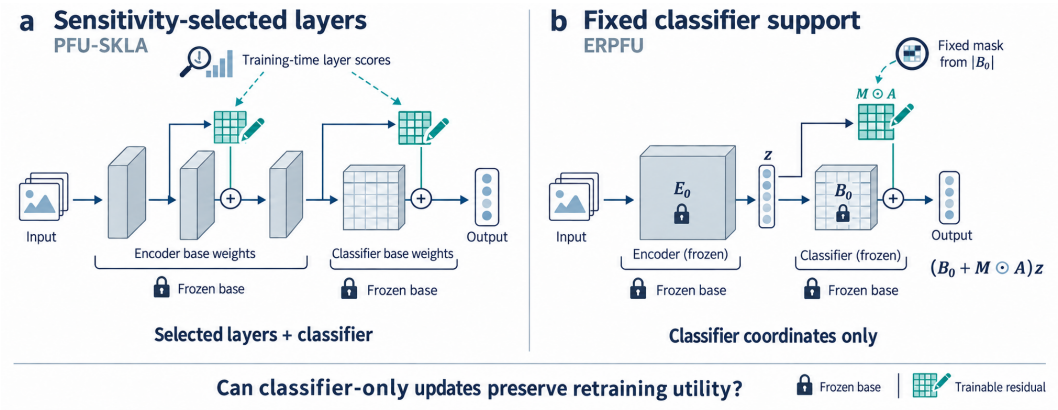


Figure 1: Adaptation support as a design choice. Left: PFU-SKLA (Lai et al., 2026) places sparse residuals at layers selected using training-time sensitivity statistics, including the classifier. Right: ERPFU confines its residual to a fixed magnitude-selected support in the final classifier, keeping the encoder unchanged. Both freeze the original base weights. Teal grids denote trainable residuals; dashed arrows denote selection. Layer and coordinate counts are schematic. The contrast motivates examining how much retraining utility a classifier-restricted correction can preserve.

chosen mask may exclude useful correction coordinates, and the frozen representation may not support the retrained predictor. Finite optimization introduces a third error. This decomposition gives a precise interpretation of when additional local training can help and when the update space itself is the limitation.

We instantiate this design in Efficient and Reversible Proactive Federated Unlearning (ERPFU). During initial federated training, weight orthogonality regularization (WOR) penalizes layer Gram residuals. At a deletion request, a sparse classifier adapter (SCA) optimizes fixed residual coordinates using only retained data. The encoder, base classifier, and feature-evaluation state remain fixed. Relative to PFU-SKLA, the change is from sensitivity-selected layers to a predetermined classifier space with an explicit support constraint. WOR supplies the preparation rule; its effect on classifier suitability is separate from the mathematical properties of the restricted objective.

The available results show where this design retains useful behavior. Across four IID client-deletion settings, ERPFU's retained accuracy is within 1.85 percentage points of retraining. In a CIFAR-10 split separating unique and shared classes, it retains 21.35% accuracy on the shared class, compared with 15.94% for retraining, while retained-only accuracy is 57.27% versus 58.56%. The non-IID results also expose a boundary: the CIFAR-10 retention gap increases to 4.66 points. We interpret these as aggregate utility measurements and report deletion-associated predictions alongside the retraining reference.

The contributions are:

- **An explicit request-time update space.** ERPFU confines retained-data adaptation to a fixed subset of classifier residual coordinates, making the trainable budget and frozen state precise.
- **A conditional account of approximation.** We separate optimization, support, and full-retraining reference errors, and relate the resulting logit bound to prediction agreement through classification margins. We also characterize the covariance and aggregation conditions underlying WOR.
- **An empirical characterization of utility and scope.** Existing client-deletion, shared-class, deletion-size, and heterogeneous-partition results identify settings with close retained accuracy and settings with larger gaps. Class and backdoor-sample results delineate the scope of this restricted design.

2 Related Work

Retraining references and federated updates. Machine unlearning studies removal of training-data influence through retraining, parameter modification, or algorithms with formal guarantees (Bourtoule et al., 2021; Graves et al., 2021; Golatkar et al., 2020; Guo et al., 2019; Izzo et al., 2021). Federated methods must additionally accommodate distributed retained data and client participation. FedEraser reconstructs a model from historical updates (Liu et al., 2021); FUKD combines update subtraction with knowledge distillation (Wu et al., 2022); and FedOSD addresses conflicts between unlearning and utility gradients (Pan et al., 2025). These approaches differ in stored information, trainable parameters, and request-time access. We use retraining as a functional reference and study a classifier-restricted update space.

Preparation and sparse adaptation. FUSED freezes the base model and learns sparse adapters in selected layers (Zhong et al., 2025). PFU-SKLA combines orthogonal preparation with sensitivity-guided critical-layer selection and sparse adaptation (Lai et al., 2026). FOUL uses a learning-to-unlearn phase and server-side gradient aggregation (Nguyen et al., 2026). ERPFU builds on this direction by fixing the adaptable layer to the classifier and selecting its residual support once by parameter magnitude. This removes sensitivity-based layer selection from the procedure and yields an affine prediction model in the active coordinates. The resulting conditional analysis identifies the price of fixing both the representation and the support. The empirical comparison with PFU-SKLA in Section 5.6 uses explicitly identified published values.

Functional behavior and privacy. Retained accuracy and deletion-associated accuracy describe predictions on selected distributions. They differ from membership inference (Carlini et al., 2022) and guarantees about the distribution of a released model. This distinction is central for a frozen-base adapter: removing the adapter restores the base predictor if both are exposed. We therefore formulate the theoretical result as functional approximation and delimit the empirical claims to the measurements reported.

3 Method

ERPFU separates training-time weight preparation from request-time optimization of a fixed classifier residual (Fig. 2). The complete federated procedure appears in Algorithm 1.

3.1 Setting

A server coordinates K clients with local datasets D_k. After T federated rounds, the global model is $M_g^T = (E^T, C^T)$, comprising an encoder and a linear classifier. Let $D = \bigcup_k D_k$, where samples retain their client identities. A request specifies a subset $D_f \subset D$ to delete, and $D_{r,k} = D_k \setminus D_f$ defines the local retained data. This covers client, class, and sample deletion. With $D_r = \bigcup_k D_{r,k}$, our reference is

$$M_g^f \approx \mathrm{Train}_{\mathcal{P}}(D_r), \tag{1}$$

where $\mathcal{P}$ specifies the architecture, initialization distribution, training schedule, and regularization used for retraining from scratch. This reference defines the target behavior; retention and deletion-related accuracy measure selected aspects of that behavior. Accuracy on D_f is interpreted relative to retraining: a retrained model may still classify deleted benign samples correctly through generalization.

3.2 Weight Orthogonality Regularization

During initial training, WOR penalizes deviations of layer weights from row orthogonality. For each convolutional or linear layer $l \in \mathcal{L}$, reshape its weights into $W_l' \in \mathbb{R}^{m_l \times n_l}$, with one row per output channel or neuron. Client k uses

$$\mathcal{L}_k^{\mathrm{WOR}} = \sum_{l \in \mathcal{L}} \| W_l' {W_l'}^{\top} - I_{m_l} \|_F, \tag{2}$$

and the data-dependent local objective

$$\mathcal{L}_k = \mathcal{L}_k^{\mathrm{CE}} + \lambda \mathcal{L}_k^{\mathrm{WOR}}. \tag{3}$$

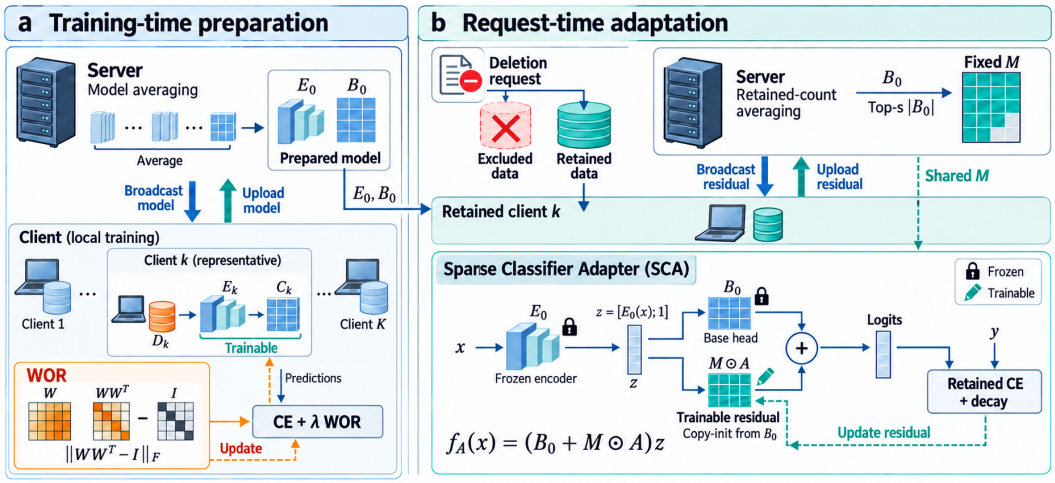


Figure 2: Overview of ERPFU. (a) Clients train local models with cross-entropy and WOR; the server averages their parameters to obtain E_0, B_0. The WOR inset illustrates the per-layer Gram penalty. (b) After a deletion request, retained data remain local. The server fixes a magnitude mask from the weights and bias of B_0. Clients freeze the encoder and base head, copy selected B_0 entries into the residual, and optimize it with retained-data cross-entropy and residual decay. Residuals are averaged using retained sample counts. Padlocks mark frozen components; the pencil marks trainable coordinates. Dashed paths denote updates or fixed-mask distribution. Matrix grids are schematic.

For the visual models, local SGD uses momentum $\beta = 0.9$ and coupled weight decay $\mu = 5 \times 10^{-4}$. Writing trainable parameters as θ, its minibatch update is

$$v_{j+1} = \beta v_j + \nabla \widehat{\mathcal{L}}_k(\theta_j) + \mu \theta_j, \qquad \theta_{j+1} = \theta_j - \eta v_{j+1}. \tag{4}$$

The optimizer is initialized with zero momentum for each client in each round. The server averages local models using their training-sample counts. Applying WOR throughout the model regularizes the weights that produce the features later reused by SCA. This is a preparation choice, not a layer-selection rule; Section 4 states the conditions under which its weight geometry controls linear-response covariance.

3.3 Sparse Classifier Adapter

At a deletion request, write the frozen encoder as $E_0 = E^T$, augment its features by $z(x) = [E_0(x); 1]$, and combine classifier weights and bias as $B_0 = [W_0, b_0] \in \mathbb{R}^{c \times (d+1)}$. Here c is the number of classes and d the feature dimension; the last column and feature are omitted if there is no bias. The encoder, base classifier, batch-normalization statistics, and dropout behavior are fixed in evaluation mode.

The server jointly ranks the absolute weight and bias entries of B_0. It selects $s = \max\{1, \mathrm{round}(\rho P)\}$ entries, where $P = c(d+1)$, and shares the resulting binary mask M across clients for the entire request. Ties use the implementation's top-s selection, and the realized mask is saved. The active fraction $\rho = 0.90$ means 90% of classifier coordinates are eligible for updates; it does not mean 90% zeros. The SCA logits are

$$f_A(x) = (B_0 + M \odot A)z(x). \tag{5}$$

Only selected residual coordinates influence predictions. The residual uses copy initialization, $M \odot A^0 = M \odot B_0$: selected entries are copied from the classifier, so their effective coefficients initially double. Unselected random entries remain masked out. A zero residual initialization would instead preserve the original function and constitutes a separate algorithm variant.

Let $u \in \mathbb{R}^s$ collect the active entries of A. For $n_{r,k} = |D_{r,k}| > 0$, the request-time objective is

$$F_k(u) = \frac{1}{n_{r,k}} \sum_{(x,y) \in D_{r,k}} \mathbb{E}_a[\ell_{\mathrm{CE}}(f_u(a(x)), y)] + \frac{\mu}{2}\|u\|_2^2. \tag{6}$$

Here a is the training augmentation (the identity when disabled). Weight decay acts on the residual, not on the effective classifier $B_0 + M \odot A$; WOR is not applied in this phase. Each participating client starts with $u_{k,0} = u^t$, $v_{k,0} = 0$, and performs local minibatch updates

$$v_{k,j+1} = \beta v_{k,j} + \nabla \widehat{L}_{r,k,j}(u_{k,j}) + \mu u_{k,j}, \qquad u_{k,j+1} = u_{k,j} - \eta v_{k,j+1}, \tag{7}$$

where $\widehat{L}_{r,k,j}$ is the minibatch cross-entropy. Let $u_k^{t,+}$ be the result after the local steps. For the participating nonempty retained clients $\mathcal{K}_t$, aggregation is

$$u^{t+1} = \sum_{k \in \mathcal{K}_t} \frac{n_{r,k}}{\sum_{q \in \mathcal{K}_t} n_{r,q}} u_k^{t,+}. \tag{8}$$

The final classifier is $B_0 + M \odot A^{T_u}$. Removing the residual restores the base model; folding it into the classifier gives the same logits in a single head. Reversibility is a property of this parameterization, rather than a guarantee about removal of information from the frozen model.

Structural budget. Let P_E be the encoder parameter count. Full-model optimization updates $P_E + P$ parameters; a dense classifier residual updates P; SCA has s effective trainable coordinates. The main restriction is to the classifier: $\rho = 0.90$ removes only 10% of its coordinates. The frozen encoder still requires a forward pass, while its gradients and optimizer updates are unnecessary. The implementation stores a dense residual tensor, so s is an effective update dimension rather than a measured payload or a guarantee of sparse-kernel acceleration. This distinction separates the algorithmic budget from the historical run times in Appendix D.

4 When Can a Fixed Classifier Approximate Retraining?

The fixed representation and mask make it possible to separate three sources of approximation error. We first analyze SCA in that space, then state what WOR's geometry does and does not control. Complete proofs are in Appendix B.

4.1 Optimization, support, and the retraining reference

Let $G(A)$ be retained-data cross-entropy on the frozen augmented features z, with residual regularization $\mu\|A\|_F^2/2$, and let $A_* = \arg\min_A G(A)$ be its dense minimizer. Inserting only the active coordinates gives $F(u) = G(J_M u)$. Both objectives are μ-strongly convex for $\mu > 0$. If retained augmented features have finite second moments, G is L-smooth; a valid choice is

$$L = \mu + |D_r|^{-1} \sum_{(x,y) \in D_r} \mathbb{E}_a \|z(a(x))\|_2^2. \tag{9}$$

This property follows from affine logits and convex cross-entropy. It concerns the classifier objective, not the full network or a convergence rate for federated local updates.

Fix a retraining realization f_{rt} under protocol $\mathcal{P}$. On a domain $\mathcal{X}$ with $\sup_{x \in \mathcal{X}} \|z(x)\|_2 \le R$, define

$$g = \|\nabla F(\widehat{u})\|_2, \qquad r_M = \|(\mathbf{1} - M) \odot A_*\|_F, \tag{10}$$
$$\varepsilon_{\mathrm{ref}} = \sup_{x \in \mathcal{X}} \|(B_0 + A_*)z(x) - f_{\mathrm{rt}}(x)\|_2.$$

Here g is the full retained-objective gradient residual, r_M is the dense correction excluded by the mask, and $\varepsilon_{\mathrm{ref}}$ compares the best regularized dense residual with full retraining. For finite $\varepsilon_{\mathrm{ref}}$, Proposition 3 gives

$$\sup_{x \in \mathcal{X}} \|f_{\widehat{u}}(x) - f_{\mathrm{rt}}(x)\|_2 \le \underbrace{\varepsilon_{\mathrm{ref}}}_{\text{reference}} + \underbrace{Rg/\mu}_{\text{optimization}} + \underbrace{R\sqrt{L/\mu}\, r_M}_{\text{support}} \equiv B. \tag{11}$$

Strong convexity controls the distance to the masked optimum by g/μ. Projecting the dense optimum onto the mask and applying smoothness bounds the support error; a triangle inequality then introduces the full-retraining reference.

Each term corresponds to a different design decision. More accurate optimization can reduce g without changing the other two terms. Activating the entire classifier removes r_M, but the reference

discrepancy may persist because the encoder is frozen and the residual regularizer differs from the retraining objective. The magnitude mask is a fixed selection rule; neither large pretrained coefficients nor a high active fraction imply a small excluded correction. This decomposition specifies the conditions for approximation without assuming they hold for every request.

4.2 From logit discrepancy to classification behavior

Let $m_{\mathrm{rt}}(x)$ be the difference between the largest and second-largest retraining logits. If $m_{\mathrm{rt}}(x) > 2B$, Eq. 11 guarantees identical top-1 predictions. More generally, on any labeled finite set $S \subseteq \mathcal{X}$,

$$|\operatorname{Acc}_S(f_{\widehat{u}}) - \operatorname{Acc}_S(f_{\mathrm{rt}})| \le D_S \le \frac{1}{|S|}\sum_{x\in S}\mathbf{1}[m_{\mathrm{rt}}(x) \le 2B], \tag{12}$$

where D_S is their prediction-disagreement fraction and accuracy is expressed in $[0, 1]$ (Corollary 4). Thus small approximation error preserves predictions away from the decision boundary. The converse is false: equal aggregate accuracy can conceal different predictions. The experiments report accuracy gaps, not estimates of B or a validation of its individual terms.

4.3 The role of weight preparation

WOR controls a weight Gram residual. Its relation to response covariance also depends on the layer inputs. For centered input covariance $\Sigma = \sigma^2 I + \Delta$, $\|\Delta\|_2 \le \delta$, and $\|WW^\top - I\|_F \le \epsilon$, Proposition 1 gives, for distinct rows,

$$|\operatorname{Cov}(w_i^\top x, w_j^\top x)| \le \sigma^2\epsilon + \delta(1+\epsilon). \tag{13}$$

The input-anisotropy term remains even for orthogonal weights. Under client averaging, $\bar{W} = \sum_k p_k W_k$, the exact identity

$$\bar{W}\bar{W}^\top - I = \sum_k p_k(W_k W_k^\top - I) - \sum_k p_k(W_k - \bar{W})(W_k - \bar{W})^\top \tag{14}$$

shows that client disagreement also changes the global Gram geometry. These results explain why local regularization does not by itself guarantee globally decorrelated responses. WOR affects the model from which A_* and $\varepsilon_{\mathrm{ref}}$ are defined; establishing that it reduces the SCA error terms requires separate evidence. The conditional analysis neither asserts this causal link nor certifies privacy erasure.

5 Experiments

We organize the available results around three questions: how much retained utility survives the classifier restriction; how this changes with data heterogeneity and deletion scope; and how the design compares with sensitivity-guided adaptation. All numerical comparisons below are descriptive point estimates.

5.1 Setup and measurements

Data and training. We use FashionMNIST (Xiao et al., 2017) with a LeNet-style CNN, CIFAR-10 (Krizhevsky et al., 2009) with ResNet-18, and CIFAR-100 with ResNet-18 or SimpleViT (Yoshioka, 2024). The setup has 20 clients, full participation, 200 initial training rounds, and one local epoch per round. IID partitions provide the primary comparison; a Dirichlet partition with $\alpha = 0.1$ tests heterogeneous retained data. The default preparation weight is $\lambda = 0.3$ and the active classifier fraction is $\rho = 0.90$. Client-deletion results use label-flipping Byzantine clients (Fung et al., 2020); sample deletion concerns injected backdoor samples. These settings evaluate targeted corrective adaptation and do not cover arbitrary benign-sample deletion.

Comparators and provenance. Retrain trains a new model on retained data; FedEraser (Liu et al., 2021), FUSED (Zhong et al., 2025), and FedOSD (Pan et al., 2025) supply reconstruction, adapter, and gradient-based comparisons. The IID baseline accuracy entries also appear in the PFU-SKLA benchmark (Lai et al., 2026); they are reused benchmark values, not newly rerun baselines. ERPFU

Table 1: IID client-deletion results (%). RA measures retained utility; FA describes the deleted-distribution test partition. Bottom rows are absolute differences from retraining, computed from the displayed values.

	FashionMNIST		CIFAR-10		CIFAR-100 R18		CIFAR-100 ViT	
Method	RA	FA	RA	FA	RA	FA	RA	FA
Retrain	99.52	0.00	61.78	2.73	32.48	0.42	31.06	0.78
FedEraser	99.31	0.00	54.29	5.23	16.57	0.54	26.06	0.78
FUSED	99.24	0.00	54.58	6.28	26.08	0.63	26.23	1.28
FedOSD	98.33	0.24	52.53	5.57	23.19	0.97	25.73	0.42
ERPFU	99.31	0.00	59.93	4.27	31.69	0.22	31.73	0.49
ERPFU Δ	0.21	0.00	1.85	1.54	0.79	0.20	0.67	0.29

Table 2: Client deletion with non-IID data ($\alpha = 0.1$). Accuracies are percentages; Δ_{RA} is the absolute RA difference from retraining.

	Retrain		ERPFU		
Dataset/backbone	RA	FA	RA	FA	Δ_{RA}
FashionMNIST	97.72	0.12	98.93	0.11	1.21
CIFAR-10 / R18	52.95	7.37	48.29	4.73	4.66
CIFAR-100 / R18	22.81	0.58	22.93	0.43	0.12
CIFAR-100 / ViT	24.72	0.21	22.93	0.44	1.79

entries are retained from the existing experiment tables. The available records contain point estimates without seed-level uncertainty. Published PFU-SKLA results are separately identified in Section 5.6; no matched-seed comparison is implied.

Metrics. RA and FA are accuracies on independent test partitions associated with retained and deleted data, respectively. They are not training-set membership measurements. We read FA against retraining, since generalization from retained data can preserve predictions on a deleted distribution. Define $\Delta_{\mathrm{RA}} = |\mathrm{RA}_u - \mathrm{RA}_{\mathrm{rt}}|$ and analogously Δ_{FA}, in percentage points. For backdoor-sample deletion, 0A is class-0 accuracy and PS is precision for predicted class 0. Historical computation times and their measurement limits are retained in Appendix D; they are not used to infer total lifecycle savings.

5.2 Retained utility under a fixed classifier update

Table 1 answers the primary utility question. ERPFU's RA differs from retraining by 0.21, 1.85, 0.79, and 0.67 points across the four settings. On the three CIFAR settings, its RA is higher than the strongest listed non-retraining baseline by 5.35, 5.61, and 5.50 points, respectively. On Fashion-MNIST it matches FedEraser at 99.31%. These observations show that the classifier restriction can preserve substantial retained utility in the evaluated client-deletion setting.

The corresponding FA differences from retraining are 0.00, 1.54, 0.20, and 0.29 points. ERPFU's CIFAR-10 FA remains above retraining, while its CIFAR-100 values fall below that reference. We report both directions because lower FA alone does not establish closer approximation. Likewise, the slightly higher RA than retraining on CIFAR-100 ViT is an observed accuracy outcome, rather than evidence of statistical superiority or more complete removal.

5.3 Heterogeneous retained data

Table 2 brings heterogeneous partitions into the main evaluation. Three settings have RA differences of at most 1.79 points, whereas CIFAR-10 has a 4.66-point deficit, larger than its 1.85-point IID deficit. The results therefore delimit the retention claim: a fixed classifier space need not preserve the same utility under a changed client distribution. The analysis allows limitations from both the frozen reference and restricted support, but these accuracy measurements do not isolate which term

Table 3: CIFAR-10 split with unique-forgotten (UF), overlapping (O), and unique-retained (UR) classes. O-Acc and UR-Acc measure preservation of different retained distributions.

Method	UF-Acc	O-Acc	UR-Acc
Retrain	0.00	15.94	58.56
FedEraser	0.00	12.39	36.97
FUSED	0.00	18.64	54.89
FedOSD	0.00	11.23	56.17
ERPFU	0.00	21.35	57.27

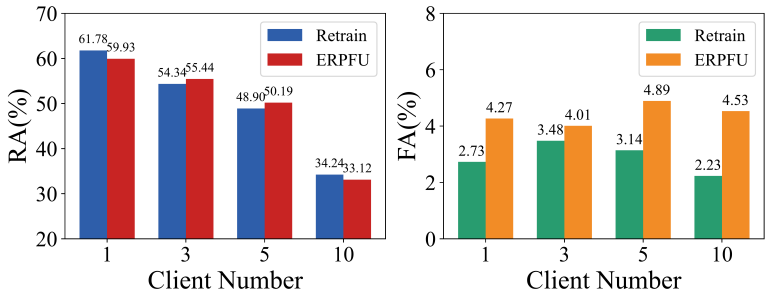


Figure 3: CIFAR-10 IID results for separate requests removing $C \in \{1, 3, 5, 10\}$ clients. RA declines with the amount of removed data; the absolute RA difference between the two methods stays within 1.85 points. FA is non-monotonic for both methods.

accounts for the observed gap. The lower CIFAR-10 FA of ERPFU (4.73% versus 7.37%) is also a difference from the target, not an independent improvement criterion.

5.4 Shared classes and deletion size

Shared-class preservation. The forget client holds all class-1 data and 90% of class-0 data in a CIFAR-10 split. Other clients retain classes 2–9 and the remaining 10% of class 0. This separates a unique-forgotten class from a class that must remain useful despite losing most of its examples. In Table 3, ERPFU obtains 21.35% shared-class accuracy, 2.71 points above FUSED and 5.41 above retraining. Its unique-retained accuracy is 57.27%, 1.29 points below retraining. All methods have zero unique-forgotten accuracy. The joint result supports preservation of shared-class predictions alongside the observed change on the unique-forgotten class; it does not establish isolation or erasure of internal representations.

Deletion size. Figure 3 varies the number of deleted clients. Retraining RA is 61.78, 54.34, 48.90, and 34.24%; ERPFU obtains 59.93, 55.44, 50.19, and 33.12%. Thus the absolute RA gaps are 1.85, 1.10, 1.29, and 1.12 points. Both methods lose retained accuracy as more clients are removed. FA ranges from 2.23–3.48% for retraining and 4.01–4.89% for ERPFU, preserving a discrepancy even when their retained accuracies are close. Each value of C defines a separate request; this comparison does not evaluate a sequence of cumulative deletions.

5.5 Other request scopes and active fractions

The class-deletion and backdoor-sample results in Appendix C characterize additional uses of the same update rule. In class deletion, ERPFU RA is 99.63, 56.53, 23.96, and 29.53%, compared with retraining at 99.21, 58.28, 24.75, and 27.49%. All compared methods have zero deleted-class accuracy. This shows preservation of retained-class utility alongside suppression of predictions on the deleted-class partition; shared information can still remain in the frozen encoder.

For backdoor-sample deletion, ERPFU's class-0 accuracy and precision are at least as high as the listed Retrain and FUSED values in all four settings. On CIFAR-10, these values are 61.85% and 70.38%. This evidence concerns class-specific behavior after a poisoned-data request. Without a

Table 4: Retained accuracy (%) in the published PFU-SKLA benchmark (†, Table 1 of Lai et al. (2026)) and the ERPFU records. This is a literature comparison under the reported nominal benchmark settings, not a new matched-run experiment.

	Client deletion		Class deletion	
Dataset/backbone	PFU-SKLA†	ERPFU	PFU-SKLA†	ERPFU
FashionMNIST	99.26	99.31	99.63	99.63
CIFAR-10 / R18	57.72	59.93	59.40	56.53
CIFAR-100 / R18	26.87	31.69	28.75	23.96
CIFAR-100 / ViT	27.18	31.73	30.25	29.53

triggered attack-success measurement or a membership audit, it does not establish backdoor eradication or privacy protection.

The available active-fraction sweep (Appendix E) reports RA between 58.25% and 59.93% for $\rho \in \{0.80, 0.85, 0.90, 0.95\}$ on CIFAR-10. The default $\rho = 0.90$ has the highest RA among these four recorded settings, but FA is non-monotonic. Because changing a copy-initialized mask also changes the initial predictor, the sweep jointly reflects support and initialization. It does not isolate the benefit of magnitude selection or the causal effect of WOR.

5.6 Comparison with sensitivity-guided adaptation

Table 4 places the fixed-classifier design next to its closest proactive predecessor. In the three CIFAR client-deletion settings, ERPFU has RA 2.21, 4.82, and 4.55 points above the published PFU-SKLA values. The ordering reverses for class deletion, where PFU-SKLA has higher RA on those same three settings. PFU-SKLA also reports higher CIFAR-10 backdoor-sample metrics (0A 71.66%, PS 76.47%) and a lower client-request time (231.77 seconds versus ERPFU's recorded 721.21 seconds). These cross-record observations support a scope-specific comparison, not universal dominance. The methodological distinction is the fixed, analyzable classifier space; expanding to sensitivity-selected layers remains a different choice of adaptable capacity.

What the evidence establishes. The tables support retained-utility comparisons in the stated settings, and the conditional results describe when a restricted classifier can approximate a retraining predictor. They provide complementary evidence: the accuracy results do not measure the bound's error terms. The available measurements do not establish WOR–SCA causal synergy, statistical significance, or membership privacy. In particular, exposing the base model alongside its residual permits exact recovery of that base by removing the residual. These limits define the scope of the present functional-adaptation claim.

6 Conclusion

ERPFU studies federated deletion requests through a fixed classifier update space. Its residual formulation makes the request-time optimization dimension explicit, while its conditional analysis separates optimization error, unsupported correction coordinates, and the gap to full retraining. Existing results show retained accuracy within 1.85 points of retraining across the four IID client-deletion settings and preserve useful shared-class predictions; heterogeneous data and comparisons with PFU-SKLA reveal scope-dependent trade-offs. This evidence supports classifier-restricted adaptation as a practical utility-preserving design in the evaluated settings. Establishing preparation-induced improvements in the error terms and removal of recoverable information remains beyond these results.

AI USE STATEMENT

Generative AI tools assisted with manuscript assessment, revision planning, mathematical derivation and proof revision, literature retrieval, evaluation-protocol and experiment-table preparation, conceptual figure generation, and LaTeX editing and formatting.

REFERENCES

Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pp. 141–159, Piscataway, NJ, USA, 2021. IEEE.

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramèr. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy*, pp. 1897–1914, 2022. doi: 10.1109/SP46214.2022.9833649.

Clement Fung, Chris JM Yoon, and Ivan Beschastnikh. The limitations of federated learning in sybil settings. In *Proceedings of the 23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID)*, pp. 301–316, Berkeley, CA, USA, 2020. USENIX Association.

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9304–9312, Los Alamitos, CA, USA, 2020. IEEE Computer Society.

Laura Graves, Vineel Nagisetty, and Vijay Ganesh. Amnesiac machine learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 11516–11524, Palo Alto, CA, USA, 2021. AAAI Press.

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

Zachary Izzo, Mary Anne Smart, Kamalika Chaudhuri, and James Zou. Approximate data deletion from machine learning models. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, pp. 2008–2016, Virtual Event, 2021. PMLR.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Jinshan Lai, Fengchun Zhang, Yunyuan Wang, Dongfen Li, Yang Zhang, Xiong Li, and Ruijin Wang. Proactive federated unlearning: Sensitivity-guided sparse adaptation on key layers. In *Proceedings of the Thirty-Fifth International Joint Conference on Artificial Intelligence*, pp. 4439–4447, 2026. doi: 10.24963/ijcai.2026/494.

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

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

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.

Zibin Pan, Zhichao Wang, Chi Li, Kaiyan Zheng, Boqi Wang, Xiaoying Tang, and Junhua Zhao. Federated unlearning with gradient descent and conflict mitigation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 19804–19812, Palo Alto, CA, USA, 2025. AAAI Press.

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

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747, 2017. arXiv:1708.07747.

Kentarou Yoshioka. vision-transformers-cifar10: Training vision transformers and related models on cifar-10. `https://github.com/kentaroy47/vision-transformers-cifar10`, 6 2024. GitHub repository.

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, Los Alamitos, CA, USA, 2025. IEEE Computer Society.

A FULL ALGORITHM

Algorithm 1 ERPFU with a fixed magnitude mask

Input: datasets $\{D_k\}_{k=1}^{K}$, deletion set D_f, rounds T, T_u, local step counts $J_{k,t}$, learning rate η, WOR coefficient λ, active fraction ρ, momentum β, weight decay $\mu > 0$
Output: logits $f(x) = (B_0 + M \odot A^{T_u})z(x)$
Stage 1: Proactive readiness
for $t = 0, \ldots, T-1$ **do**
  Broadcast global parameters to participating clients
  Each client resets momentum and takes local steps by Eq. 4, using Eq. 3
  Average client models with weights proportional to local sample counts
end for
Obtain E_0, B_0; freeze all base parameters and evaluation-state buffers
Stage 2: Fixed-support adaptation
Form $D_{r,k} = D_k \setminus D_f$; exclude clients with $|D_{r,k}| = 0$
Select a shared top-s magnitude mask M from weight and bias jointly
Initialize active residual coordinates from B_0; keep M fixed
for $t = 0, \ldots, T_u - 1$ **do**
  Broadcast u^t to participating retained clients $\mathcal{K}_t$
  for each $k \in \mathcal{K}_t$ **do**
    Set $u_{k,0} = u^t$, $v_{k,0} = 0$
    for $j = 0, \ldots, J_{k,t} - 1$ **do**
      Draw a minibatch from $D_{r,k}$; compute logits by Eq. 5
      Update the active residual by Eq. 7
    end for
    Return $u_k^{t,+} = u_{k,J_{k,t}}$
  end for
  Aggregate by Eq. 8 using retained sample counts
end for
Insert u^{T_u} into the fixed support to obtain $M \odot A^{T_u}$

The specification uses the retained examples supplied to local training. If a protocol further subsamples those examples, its aggregation counts and optimization reference must describe that subsampling. The visual implementation uses $\beta = 0.9$, $\mu = 5 \times 10^{-4}$, and recreates each client's optimizer every round. The mathematical vector u describes effective trainable coordinates; it does not assert that the implementation packs only those coordinates into transmitted tensors.

B Conditional Analysis of Readiness and Adaptation

We analyze three quantities: linear-response covariance, weight geometry after aggregation, and the discrepancy between SCA and a specified retraining reference. The first two concern WOR; the third identifies the conditions under which a fixed classifier mask can approximate retraining. All statements concern the SCA definition in Section 3.3, including its fixed magnitude mask and residual regularization.

B.1 Weight geometry and input covariance

Let $W \in \mathbb{R}^{m\times n}$ have rows $w_i^\top$, and let a centered layer input x have covariance Σ. Its linear responses are $h_i(x) = w_i^\top x$. Weight orthogonality alone does not imply decorrelated responses: for $W = I_2$ and

$$\Sigma = \begin{pmatrix} 1 & 1/2 \\ 1/2 & 1 \end{pmatrix},$$

the Gram residual is zero while $\mathrm{Cov}(h_1, h_2) = 1/2$. In particular, the inequality $|w_i^\top \Sigma w_j| \le \|\Sigma\|_2 |w_i^\top w_j|$ is false in general.

Proposition 1 (Covariance with an anisotropy term) *Suppose $\|WW^\top - I_m\|_F \le \epsilon$, and write $\Sigma = \sigma^2 I_n + \Delta$, where $\sigma^2 \ge 0$ and $\|\Delta\|_2 \le \delta$. For every $i \ne j$,*

$$|\,\mathrm{Cov}(h_i(x), h_j(x))| \le \sigma^2\epsilon + \delta\|w_i\|_2\|w_j\|_2 \le \sigma^2\epsilon + \delta(1+\epsilon). \tag{15}$$

Proof. Centering gives $\mathrm{Cov}(h_i, h_j) = w_i^\top \Sigma w_j$. The Gram bound implies $|w_i^\top w_j| \le \epsilon$ for $i \ne j$ and $\|w_i\|_2^2 \le 1 + \epsilon$. Substituting the covariance decomposition and using the spectral norm inequality yields

$$|w_i^\top \Sigma w_j| \le \sigma^2 |w_i^\top w_j| + |w_i^\top \Delta w_j| \le \sigma^2\epsilon + \delta\|w_i\|_2\|w_j\|_2.$$

The row-norm bounds give the last inequality in Eq. 15. □

The anisotropy term is essential. For example, choosing $\sigma^2 = \mathrm{tr}(\Sigma)/n$ makes $\delta = \|\Sigma - \sigma^2 I_n\|_2$ directly measurable. Small covariance requires controlling both this term and the Gram residual. Batch normalization alone does not establish a small δ. The proposition concerns linear preactivations, not independence after nonlinear transformations or separation of client-specific information. If $m > n$, the rank constraint gives $\|WW^\top - I_m\|_F \ge \sqrt{m-n}$, because at least $m-n$ eigenvalues of $WW^\top$ are zero.

B.2 Aggregation and client disagreement

Proposition 2 (Gram residual under weight averaging) *Let W_k be equally shaped client weight matrices in the same parameter coordinates, with $p_k \ge 0$, $\sum_k p_k = 1$, and $\bar{W} = \sum_k p_k W_k$. Then*

$$\bar{W}\bar{W}^\top - I = \sum_k p_k (W_k W_k^\top - I) - \sum_k p_k (W_k - \bar{W})(W_k - \bar{W})^\top, \tag{16}$$

and consequently

$$\|\bar{W}\bar{W}^\top - I\|_F \le \sum_k p_k \|W_k W_k^\top - I\|_F + \sum_k p_k \|W_k - \bar{W}\|_F^2. \tag{17}$$

Proof. Expanding the second sum in Eq. 16 gives $\sum_k p_k W_k W_k^\top - \bar{W}\bar{W}^\top$, since $\sum_k p_k W_k = \bar{W}$. Rearrangement proves the identity. The triangle inequality and $\|HH^\top\|_F \le \|H\|_F^2$, which follows by summing fourth powers of the singular values of H, give Eq. 17. □

Thus, local Gram residuals and client disagreement jointly control this bound. WOR penalizes the former; preservation after FedAvg also depends on the latter. This identity does not assume that disagreement is small or that feature covariances align across clients.

B.3 A DIAGNOSTIC BOUND FOR FIXED-SUPPORT ADAPTATION

Let E_0, B_0, $z(x)$, and M be the frozen encoder, base classifier, augmented features, and fixed mask defined in Section 3.3. Define a *dense residual reference* on the same frozen model:

$$G(A) = \frac{1}{N_r} \sum_{(x,y)\in D_r} \mathbb{E}_a[\ell_{\mathrm{CE}}((B_0 + A)z(a(x)), y)] + \frac{\mu}{2}\|A\|_F^2, \qquad A_* = \arg\min_A G(A), \quad (18)$$

where $N_r = |D_r| > 0$ and $\mu > 0$. Let $J_M : \mathbb{R}^s \to \mathbb{R}^{c\times(d+1)}$ insert the active coordinates into the positions selected by M, setting all other entries to zero. Thus $\|J_M u\|_F = \|u\|_2$, and the SCA objective is $F(u) = G(J_M u)$.

Both objectives are μ-strongly convex: logits are affine in trainable parameters, cross-entropy is convex in logits, and the quadratic regularizer covers every trainable coordinate. Suppose retained augmented features have finite second moments. Then G is globally L-smooth; one valid choice is $L = \mu + N_r^{-1}\sum_{(x,y)\in D_r} \mathbb{E}_a\|z(a(x))\|_2^2$. The logit Hessian is $\mathrm{diag}(p) - pp^\top$, whose quadratic form is a variance bounded by the squared norm of its argument. Composition with the feature map and averaging give this bound. These properties establish unique minimizers of the frozen-feature objectives, not convergence of local momentum/FedAvg iterations.

To compare with full retraining, fix its protocol $\mathcal{P}$ and a realized model trained on D_r from scratch, with logits $f_{\mathrm{rt}}(x)$ in the same class order. On an evaluation domain $\mathcal{X}$, define

$$r_M = \|(\mathbf{1} - M) \odot A_*\|_F, \qquad \varepsilon_{\mathrm{ref}} = \sup_{x\in\mathcal{X}} \|(B_0 + A_*)z(x) - f_{\mathrm{rt}}(x)\|_2. \quad (19)$$

Here r_M measures the dense correction excluded by the mask. The reference discrepancy includes the frozen representation, the regularizer centered at B_0, and differences from protocol $\mathcal{P}$. The dense reference is therefore distinct from full retraining.

Proposition 3 (Optimization, support, and reference errors) *Under the conditions above, assume* $\sup_{x\in\mathcal{X}} \|z(x)\|_2 \le R < \infty$ *and* $\varepsilon_{\mathrm{ref}} < \infty$. *For any returned SCA iterate* $\widehat{u}$, *let* $g = \|\nabla F(\widehat{u})\|_2$, *evaluated on the full retained objective including regularization. Then*

$$\sup_{x\in\mathcal{X}} \|f_{\widehat{u}}(x) - f_{\mathrm{rt}}(x)\|_2 \le \varepsilon_{\mathrm{ref}} + R\left(\frac{g}{\mu} + \sqrt{\frac{L}{\mu}}\, r_M\right). \quad (20)$$

Proof. Let $u_* = \arg\min_u F(u)$ and $A_M = J_M u_*$. First, strong convexity and $\nabla F(u_*) = 0$ imply

$$\mu\|\widehat{u} - u_*\|_2^2 \le \langle \nabla F(\widehat{u}), \widehat{u} - u_*\rangle \le g\|\widehat{u} - u_*\|_2.$$

If the difference is zero the desired bound holds; otherwise division gives $\|\widehat{u} - u_*\|_2 \le g/\mu$. Second, $M \odot A_*$ is feasible for the masked problem. Optimality of A_M, smoothness of G, and $\nabla G(A_*) = 0$ give

$$G(A_M) - G(A_*) \le G(M \odot A_*) - G(A_*) \le \frac{L}{2} r_M^2.$$

Strong convexity of G gives the complementary lower bound $G(A_M) - G(A_*) \ge (\mu/2)\|A_M - A_*\|_F^2$. Hence $\|A_M - A_*\|_F \le \sqrt{L/\mu}\, r_M$. Finally, for every $x \in \mathcal{X}$,

$$\begin{aligned}\|f_{\widehat{u}}(x) - f_{\mathrm{rt}}(x)\|_2 &\le \|(J_M\widehat{u} - A_*)z(x)\|_2 + \varepsilon_{\mathrm{ref}} \\ &\le R\big(\|\widehat{u} - u_*\|_2 + \|A_M - A_*\|_F\big) + \varepsilon_{\mathrm{ref}}.\end{aligned}$$

Substitution proves Eq. 20. □

Interpretation and measurable quantities. The bound applies to any iterate, including one produced by local momentum and FedAvg, through its actual full-objective residual g; it is not a convergence-rate statement. Even exact optimization only removes this term. The support error and reference discrepancy can remain large. With all coordinates active, $r_M = 0$; with $A_* = 0$, it is zero for every mask. The coefficient $1/\mu$ may make the bound loose for small weight decay.

For $A_* \neq 0$, the selected correction-energy fraction is $q_M = \|M \odot A_*\|_F^2/\|A_*\|_F^2$. Conditional on A_*, a uniformly random mask of size s has expected $q_M = s/P$. Thus the active fraction alone does not establish a selection advantage, and WOR's effect on these quantities is not a premise of Proposition 3.

Corollary 4 (Prediction agreement away from the boundary) *Under Proposition 3, let B denote the right-hand side of Eq. 20. For the retraining model's top-1 logit margin $m_{\mathrm{rt}}(x)$, $m_{\mathrm{rt}}(x) > 2B$ implies identical top-1 predictions from $f_{\widehat{u}}$ and f_{rt}. On any finite labeled evaluation set $S \subseteq \mathcal{X}$, their accuracy difference and prediction-disagreement fraction satisfy Eq. 12.*

Proof. The logit ℓ_2 bound implies that each coordinate changes by at most B. If j is the unique largest retraining logit, then for every $k \neq j$,

$$f_{\widehat{u},j}(x) - f_{\widehat{u},k}(x) \geq f_{\mathrm{rt},j}(x) - f_{\mathrm{rt},k}(x) - 2B \geq m_{\mathrm{rt}}(x) - 2B.$$

A margin larger than $2B$ makes this positive. Hence any disagreement must occur where $m_{\mathrm{rt}}(x) \leq 2B$, including possible ties. Finally, two correctness indicators can differ only if the predicted labels differ. Averaging their absolute difference and using the triangle inequality proves Eq. 12. □

This corollary supplies a sufficient condition for prediction agreement. A small observed accuracy difference cannot be inverted to conclude that B or the disagreement fraction is small.

Scope of the result. Equation 20 bounds logits for a fixed retraining realization on the stated domain. It does not establish indistinguishability between distributions of released models or certified privacy removal. The encoder, base classifier, and mask were obtained from the original data and can retain dependence on the forget set. Retained-loss minimization and small prediction discrepancy therefore require separate retraining comparisons and privacy audits. Folding the residual into the classifier removes the detachable adapter representation but does not itself establish removal of that dependence.

C Additional Request Scopes

Table 5 preserves the recorded IID accuracy and run-time entries for client, class, and backdoor-sample deletion. Client results are also organized around the retraining reference in the main text. For class deletion, RA measures retained-class test accuracy and FA measures deleted-class test accuracy. For backdoor-sample deletion, 0A and PS concern class 0 rather than an attack-success or privacy metric. No entries in this table are additional experiments.

D Recorded Run Times and Cost Scope

The Comp entries in Table 5 are historical run times in seconds. Their timer boundaries and preparation costs are not fully specified in the available records, so they do not support a controlled end-to-end speedup or an amortized lifecycle-cost estimate. WOR is evaluated during initial training and adds preparation work. SCA still evaluates the encoder in each forward pass even though the encoder receives no optimization updates.

For the deletion-size comparison, the recorded times are:

Deleted clients C	Retrain	ERPFU
1	9784.29	1344.67
3	6415.47	716.11
5	5911.97	625.38
10	4516.71	365.58

The $C = 1$ ERPFU time differs from the 721.21 seconds in the main IID record. The available records do not resolve that difference; neither value is used for a speedup claim. Requests at different C are separate settings rather than a sequential-deletion cost trace.

For a head with $P = c(d + 1)$ coordinates, the effective SCA update dimension is $s = \max\{1, \mathrm{round}(\rho P)\}$. A packed active-coordinate upload would use bs value bytes for b bytes per value, before metadata, while a dense residual uses bP. The implementation's dense storage does not establish packed communication. Initial model and mask distribution, uplink and downlink, optimizer state, and the number of rounds must be specified to compare actual communication.

Table 5: Complete recorded IID results across deletion scopes. The first two rows of each block are RA and FA for client/class deletion, and 0A and PS for backdoor-sample deletion. Accuracy values are percentages; Comp is seconds. FA is descriptive and is compared with retraining. Run times retain their original measurement scope.

	Client Unlearning					Class Unlearning			Sample Unlearning		
	Retrain	Fed Eraser	FUSED	Fed OSD	ERPFU	Retrain	FUSED	ERPFU	Retrain	FUSED	ERPFU
FashionMNIST based LeNet											
RA / 0A	99.52	99.31	99.24	98.33	99.31	99.21	99.02	99.63	100.00	100.00	100.00
FA / PS	0.00	0.00	0.00	0.24	0.00	0.00	0.00	0.00	93.68	93.68	94.74
Comp	882.12	1587.95	234.64	220.73	215.68	726.44	564.57	502.90	1476.22	1236.87	1184.44
CIFAR-10 based ResNet18											
RA / 0A	61.78	54.29	54.58	52.53	59.93	58.28	52.35	56.53	53.33	58.14	61.85
FA / PS	2.73	5.23	6.28	5.57	4.27	0.00	0.00	0.00	67.75	64.62	70.38
Comp	9784.30	1749.44	1787.99	811.43	721.21	5453.17	1020.24	819.22	5510.64	2170.50	1470.29
CIFAR-100 based ResNet18											
RA / 0A	32.48	16.57	26.08	23.19	31.69	24.75	21.97	23.96	50.00	75.00	80.00
FA / PS	0.42	0.54	0.63	0.97	0.22	0.00	0.00	0.00	62.00	80.00	85.00
Comp	13006.63	1733.84	1739.35	787.55	651.00	2583.95	933.39	442.84	4421.97	2931.76	2862.06
CIFAR-100 based SimpleViT											
RA / 0A	31.06	26.06	26.23	25.73	31.73	27.49	20.94	29.53	50.00	50.00	75.00
FA / PS	0.78	0.78	1.28	0.42	0.49	0.00	0.00	0.00	50.00	50.00	62.50
Comp	5653.37	5653.37	833.50	779.77	522.78	2461.29	1075.32	756.03	4436.85	2912.11	2536.76

E Classifier Active-Fraction Results

Table 6 evaluates the active-coordinate fraction ρ on CIFAR-10 client unlearning. A larger ρ selects more weight and bias entries for the fixed magnitude mask. Selected residuals copy the base classifier; unselected random entries remain masked out and have no effect on predictions.

Table 6 records the available CIFAR-10 client-deletion sweep. The default $\rho = 0.90$ has RA of 59.93%, compared with 59.40% at $\rho = 0.95$. FA varies non-monotonically. The run times are observations under the original timing procedure: a larger active support does not imply fewer operations, and this table does not establish a causal reduction in runtime from adding coordinates.

Table 6: Effect of the SCA active-coordinate fraction ρ on CIFAR-10 client unlearning.

ρ	RA ↑	FA	Comp ↓
0.95	59.40	5.59	669.43
0.90	59.93	4.27	721.21
0.85	58.25	5.45	745.51
0.80	58.27	4.69	785.05

F Measurement Scope and Reproducibility Limits

The reported values are aggregate historical records. Seed-level outputs, independent retraining variability, and complete run configurations are unavailable in this revision. The formal SCA specification describes the unified algorithm, while its exact correspondence to every historical run remains unverified. Thus the tables provide descriptive evidence, not independently reproduced estimates of the specified implementation.

Deleted-distribution test examples are distinct from the deleted training examples. A membership evaluation would require stable example identities and matched never-member data, with attack fitting and threshold calibration independent of the audit set (Carlini et al., 2022). Class or client

labels cannot substitute for membership labels. The earlier membership and recovery summaries do not provide a validated audit under this definition and are not used as evidence here.

The frozen encoder may retain information about the deleted data. Releasing a separate base model and adapter allows a weight-access adversary to recover the base by removing the adapter; folding the residual into the classifier preserves the adapted function but does not establish encoder erasure. Black-box accuracy comparisons and weight-access recovery therefore concern distinct questions. The retained-utility claim is limited to the recorded task behavior.