Proactive Federated Unlearning: Sensitivity-Guided Sparse Adaptation on Key Layers

Jinshan Lai[1], **Fengchun Zhang**[1], **Yunyuan Wang**[2], **Dongfen Li**[3], **Yang Zhang**[4], **Xiong Li**[1] and **Ruijin Wang**[1,*]

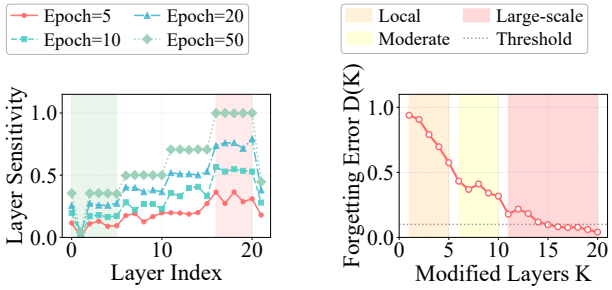
[1]University of Science and Technology of China
[2]Shanghai Jiao Tong University
[3]Chengdu University of Technology
[4]Chengdu Civil Aviation Information Technology Co., Ltd.
202311090910@std.uestc.edu.cn, {2062794989, 1700250658}@qq.com, lidongfen17@cdut.edu.cn, 719481842@qq.com, lixiongzhq@163.com, ruijinwang@uestc.edu.cn

Abstract

Driven by privacy regulations, federated unlearning (FU) aims to remove the influence of specific clients or samples from a trained federated model, approximating the behavior of retraining from scratch without the target data. However, existing FU methods are largely reactive: retraining-based solutions are accurate but prohibitively expensive, while parameter-based approaches are more efficient yet may cause irreversible knowledge damage and catastrophic forgetting. We introduce PFU-SKLA, a proactive FU framework that endows models with built-in forgettability. During pretraining, we perform Orthogonality-guided Representation Disentanglement (ORD) to learn a robust, disentangled feature space that reduces cross-client/class interference. Additionally, we use Dynamic Sensitivity-based Key Layer Identification (DS-KLI) to identify sensitive layers that hold target knowledge. During unlearning, we propose a sparse, lightweight adaptation strategy that precisely erases target knowledge by inserting sparse adapters into the identified critical layers while preserving non-target knowledge by freezing the backbone. Extensive experiments across multiple datasets and three standard unlearning settings show that PFU-SKLA consistently approaches retraining-from-scratch performance, while substantially reducing communication and computation costs compared to state-of-the-art (SOTA) FU methods.

1 Introduction

With increasing privacy requirements, Federated Learning (FL) enables multiple clients to collaboratively train a global model while keeping data local, making it attractive in domains such as healthcare, finance, and mobile services [McMahan *et al.*, 2017; Ren *et al.*, 2024]. Meanwhile, regulations such as the General Data Protection Regulation (GDPR) and the "right to be forgotten" motivate the ability to remove the influence of specific clients, classes, or samples from an already trained federated model [Regulation, 2018]. This problem, known as Federated Unlearning (FU), aims to erase target data influence without full retraining [Wu *et al.*, 2022b; Liu *et al.*, 2022; Liu *et al.*, 2024].



(a) Sensitivity of different layers.

(b) The cost of forgetting.

Figure 1: Toy experiments on CIFAR-100. (a) Layer-wise sensitivity S_l measured at different training epochs, illustrating non-uniform sensitivity across layers and its increasing trend in deeper layers. (b) Forgetting error $D(K)$ versus the number of modified layers K during post-hoc unlearning, showing that small/local modifications often fail to reach an acceptable forgetting threshold, while stronger forgetting typically requires modifying more layers.

Most existing FU methods are reactive, in the sense that unlearning is performed only after a request arrives [Liu *et al.*, 2024; Zhao *et al.*, 2025]. Prior work can be broadly grouped into several families. Update- or influence-based methods attempt to offset target contributions via gradient reversal, compensation, or approximation, which can be efficient but may rely on accurate update estimation and non-trivial correction [Halimi *et al.*, 2022; Zhao *et al.*, 2023; Khalil *et al.*, 2025; Pan *et al.*, 2025]. Retraining- or reconstruction-based methods more directly approximate a "never-learned" model, but are often expensive in federated settings with strict latency and resource constraints [Liu *et al.*, 2021; Liu *et al.*, 2022;

Che *et al.*, 2023]. More recently, hybrid or structure-based designs, such as constrained parameterization and adapter-style editing, have been explored to balance effectiveness and efficiency, although precisely localizing target knowledge can remain challenging under heterogeneous client distributions [Wu *et al.*, 2022a; Zhong *et al.*, 2025; Gao *et al.*, 2022]. A practical mismatch therefore arises: unlearning requests typically occur after training, yet most training pipelines do not explicitly organize knowledge for selective removal, so post-hoc unlearning may require broader updates or additional overhead [Liu *et al.*, 2024; Li *et al.*, 2025].

To better understand the locality–cost trade-off in post-hoc unlearning, we conduct toy experiments on CIFAR-10/100 with a layer-wise sensitivity analysis. As shown in Figure 1(a), the sensitivity score S_l is highly non-uniform across layers and evolves over training: deeper layers become increasingly sensitive as optimization proceeds, while shallow layers remain relatively stable [Lee *et al.*, 2019]. This trend suggests that later layers progressively capture more class- and client-discriminative information, whereas earlier layers preserve more generic representations. Figure 1(b) shows that when updates are restricted to only a few layers (local correction), the forgetting error $D(K)$ often stays above an acceptable threshold; reducing $D(K)$ typically requires modifying a larger set of layers [Tarun *et al.*, 2023]. This observation suggests that effective post-hoc forgetting under strict locality constraints can be difficult, and expanding the update scope increases computation and communication and may affect stability in long-running FL systems [Wu *et al.*, 2022b; Gu *et al.*, 2024].

Building on these observations, we focus on a budgeted setting where only a limited portion of the model can be modified during unlearning [Chowdhury *et al.*, 2024; Li *et al.*, 2024]. When the modification budget is small, post-hoc unlearning may not achieve sufficient forgetting; when the budget is expanded, the system cost increases. This motivates a proactive alternative: organize representations during training and track where unlearning-relevant information concentrates, so that future unlearning can be carried out within a small, well-identified subset of layers [Cao and Yang, 2015; Bourtoule *et al.*, 2021; Li *et al.*, 2025].

Motivated by this idea, we propose PFU-SKLA, a proactive FU framework guided by layer-wise sensitivity and sparse adaptation. PFU-SKLA has two phases. In the first phase, Proactive Federated Unlearning Readiness (PFUR), the model is trained with Orthogonality-guided Representation Decoupling (ORD) to reduce representation coupling [Ranasinghe *et al.*, 2021], and the server periodically estimates layer-wise sensitivity using lightweight statistics. Here, sensitivity is a training-time signal of each layer's responsiveness to data and representation constraints; ORD does not define sensitivity, but can stabilize sensitivity estimates by reducing representation entanglement. Based on these signals, PFU-SKLA selects a compact set of critical layers via Dynamic Sensitivity-based Key Layer Identification (DS-KLI). In the second phase, when an unlearning request arrives, PFU-SKLA performs unlearning via sparse critical-layer adapters. The backbone parameters are frozen, and only lightweight, sparsely parameterized adapters are inserted into the selected critical layers and optimized using the retained data in a federated manner. Restricting updates to a small number of sparse parameters enables efficient and localized unlearning, while allowing exact rollback in parameter space by removing the adapters. The remaining layers stay untouched, helping preserve performance on retained data.

In summary, our main contributions are as follows:

- We propose PFU-SKLA, a proactive federated unlearning (FU) framework that prepares models during training for efficient, localized, and controllable unlearning at deployment.
- We develop two key components: (i) ORD to reduce representation coupling and sharpen unlearning-relevant signals, and (ii) DS-KLI to select a compact set of critical layers, enabling sparse adapter-based updates with exact rollback by adapter removal.
- Extensive experiments on FashionMNIST and CIFAR-10/100 with CNN and ViT backbones, across client/class/sample unlearning and IID/non-IID settings, demonstrate competitive retention–forgetting trade-offs while substantially reducing computation and communication compared with representative baselines.

2 Related Work

2.1 Machine Unlearning

Machine Unlearning (MU) aims to remove the influence of specific training data from a trained model without full retraining, in order to meet privacy, compliance, and fairness requirements [Bourtoule *et al.*, 2021; Koundinya Gundavarapu *et al.*, 2024; Li and Sui, 2025; Liu *et al.*, 2025; Zhang *et al.*, 2023]. In centralized settings, existing MU approaches can be broadly categorized into data-level and model-level methods [Mercuri *et al.*, 2022; Nguyen *et al.*, 2025; Li *et al.*, 2025]. Data-level methods operate by modifying or reorganizing the training data, such as through data mixing, substitution, or partial retraining [Zhang *et al.*, 2022; Graves *et al.*, 2021; Tarun *et al.*, 2023; Eisenhofer *et al.*, 2025]. While computationally efficient, these approaches offer limited control over where and how knowledge is removed, and do not naturally support built-in forgettability or reversibility, making them less suitable for federated and proactive unlearning scenarios. Model-level methods directly manipulate model parameters via pruning, parameter correction, or sub-model replacement to erase target knowledge [Chowdhury *et al.*, 2024; Li *et al.*, 2024; Warnecke *et al.*, 2021]. Although more precise, they are typically reactive, may incur high computational costs, and risk catastrophic forgetting when large parameter subsets are modified. Several theoretically grounded frameworks, such as PAC-Bayesian variational unlearning and cryptographic verification, further improve unlearning guarantees [Jose and Simeone, 2021; Eisenhofer *et al.*, 2025], but are mainly designed for centralized settings.

2.2 Federated Unlearning

Federated Unlearning (FU) focuses on removing the influence of specific clients, classes, or samples from a fed-

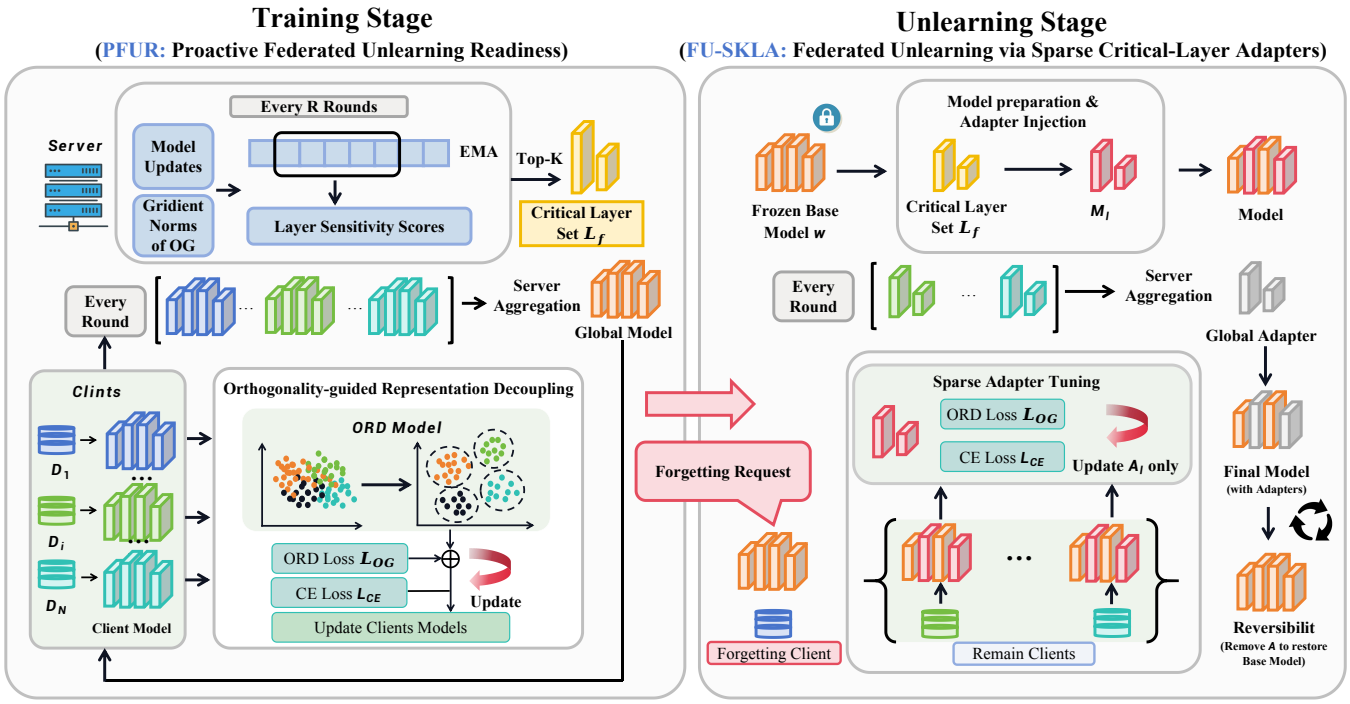


Figure 2: The PFU-SKLA framework with two stages. **Left (PFUR).** During federated training, clients update the backbone with ORD regularization to reduce feature coupling, while DS-KLI identifies a small set of critical layers $\mathcal{L}_f$ for future unlearning. **Right (FU-SKLA).** When an unlearning request arrives, the backbone is frozen and only sparse additive adapters on $\mathcal{L}_f$ are updated and aggregated under partial participation. Removing the adapters restores the pre-unlearning model and provides exact rollback.

erated model while preserving its utility on the remaining data [Liu *et al.*, 2021; Liu *et al.*, 2022; Su and Li, 2023; Wu *et al.*, 2022a; Halimi *et al.*, 2022; Zhao *et al.*, 2023; Khalil *et al.*, 2025; Deng *et al.*, 2024; Pan *et al.*, 2025]. Compared with centralized unlearning, FU introduces additional challenges due to distributed data ownership, limited client participation, and strict constraints on communication and computation [Liu *et al.*, 2024; Zhao *et al.*, 2025]. As a result, existing FU methods often face a trade-off between unlearning effectiveness and system efficiency.

Passive Federated Unlearning. Most existing FU methods adopt a passive paradigm, where unlearning is triggered only after a request is issued. These approaches are well motivated in practice, as unlearning requests are typically sporadic and unpredictable, and retraining from scratch is often infeasible. Representative methods include FedEraser [Liu *et al.*, 2021], which leverages stored historical updates for efficient model reconstruction, FUKD [Wu *et al.*, 2022a], which removes client influence via update subtraction and knowledge distillation, and NoT [Khalil *et al.*, 2025] and FedOSD [Pan *et al.*, 2025], which mitigate forgetting and optimization conflicts through parameter manipulation and optimization strategies. Despite their effectiveness, passive FU methods generally require either broad parameter updates or access to auxiliary server-side information, leading to high communication and computation costs and limited controllability. Moreover, since no structural preparation is made during training, these methods often struggle to localize and erase target knowledge without affecting retained information, especially in long-running and dynamic FL systems.

Reversible Federated Unlearning. Recent efforts have explored reversible FU to mitigate irreversible forgetting. The FUSED method introduces sparse adapters to overwrite existing knowledge, enabling rollback by removing the adapters [Zhong *et al.*, 2025]. While promising, FUSED identifies critical layers primarily based on global parameter variations, which may fail to capture locally sensitive or client-specific knowledge distributions in heterogeneous federated settings. This can lead to less precise unlearning and unnecessary updates to non-critical layers.

3 Methodology

As illustrated in Figure 2, PFU-SKLA adopts a two-stage design that separates unlearning preparation from unlearning execution. This separation is essential because effective FU requires sensitive knowledge to be structurally identifiable during training, rather than relying on costly post-hoc corrections. In the first stage, termed Proactive Federated Unlearning Readiness (PFUR), the model is trained to disentangle representations and track layer-wise sensitivity signals, enabling the localization of class- and client-specific information into a compact set of critical layers. In the second stage, unlearning is carried out by attaching sparse adapters only to these critical layers while freezing all original parameters, ensuring that updates remain lightweight, localized, and fully reversible by removing the adapters, with minimal interference to retained knowledge. Notably, proactive refers

exclusively to training-time structural or-ganization, not preemptive erasure.

3.1 Proactive Federated Unlearning Readiness

PFUR consists of two core components: orthogonality-guided representation decoupling (ORD) and periodic critical-layer updates (PCU).

Orthogonality-Guided Representation Decoupling

To facilitate more reliable identification of layers that support representation decoupling, we introduce ORD during training. By reducing representation entanglement, the orthogonality-guided (OG) constraint sharpens layer-wise sensitivity signals and leads to more stable critical-layer identification.

Assume client i with dataset $D_i = \{(x_{i,j}, y_{i,j})\}_{j=1}^{|D_i|}$ and local model parameter $w_i = (\phi_i, v_i)$, where ϕ_i is the feature extractor and v_i is the classifier. We utilize phi_i to get feature representations $f_i = \{f_{i,j}\}_{j=1}^{|D_i|} = \{f_{\theta_i}(x_{i,j})\}_{j=1}^{|D_i|}$. Then we perform ℓ_2 normalization for f_i as follows:

$$\hat{f}_i = \frac{f_i}{\|f_i\|_2}, \tag{1}$$

and construct a client-local similarity matrix over normalized features for computing the OG regularization:

$$S_i^{j,k} = \hat{f}_{i,j}^{\top} \hat{f}_{i,k}, \quad 1 \le j,k \le |D_i|. \tag{2}$$

Let $\mathcal{P}_i = \{(j,k) \mid y_{i,j} = y_{i,k},\, j \neq k\}$ denote the sets of positive pairs and $\mathcal{N}_i = \{(j,k) \mid y_{i,j} \neq y_{i,k}\}$ denote the sets of negative pairs, respectively. We define the mean similarities as follows:

$$\overline{s}_{i,\text{pos}} = \frac{1}{|\mathcal{P}_i|} \sum_{(j,k)\in\mathcal{P}_i} S_i^{j,k}, \qquad \overline{s}_{i,\text{neg}} = \frac{1}{|\mathcal{N}_i|} \sum_{(j,k)\in\mathcal{N}_i} S_i^{j,k}, \tag{3}$$

where $\overline{s}_{i,\text{pos}}$ is the positive similarity and $\overline{s}_{i,\text{neg}}$ is the negative similarity. Then we get the OG constraint as follows:

$$\mathcal{L}_{i,\text{OG}} = \big(1 - \overline{s}_{i,\text{pos}}\big) + \gamma\, \overline{s}_{i,\text{neg}}, \tag{4}$$

where $\gamma > 0$ balances the penalties on positive and negative pairs. Minimizing $\mathcal{L}_{\text{OG}}$ drives same-class features to be tightly clustered ($\overline{s}_{\text{pos}} \to 1$) and different-class features to be nearly orthogonal ($\overline{s}_{\text{neg}} \to 0$), thereby enforcing explicit inter-class separation and intra-class compactness in the feature space. We integrate this OG regularizer with the task objective $\mathcal{L}_{i,\text{CE}}$ as follows:

$$\mathcal{L}_{i,\text{total}} = \mathcal{L}_{i,\text{CE}} + \lambda \cdot \mathcal{L}_{i,\text{OPL}}, \tag{5}$$

where λ controls the strength of the OPL regularization. A single local gradient descent step at iteration t is given by

$$w_i^{t+1} = w_i^t - \eta_t\, \nabla_{w_i^t} \mathcal{L}_{i,\text{total}}\big(D_i; w_i^t\big), \tag{6}$$

where η_t is the local learning rate. These updated local models are aggregated on the server to get w^t via FedAvg.

Through its OG constraint, ORD reshapes the representation space during federated training, reducing cross-class interference while guiding the periodic critical-layer updates.

Periodic Critical-Layer Updates

During federated training, the server periodically updates the critical-layer set using layer-wise parameter update magnitudes and aggregated norms of the OG gradients, which are computed locally by clients and communicated only as lightweight statistics. Specifically, for each layer l, the parameter update and OG gradient norm are as follows:

$$\text{Diff}_l^{(\text{cu})} = \frac{1}{N} \sum_{i=1}^{N} \|w_{i,l} - w_l\|, \tag{7}$$

$$\text{GradNorm}_l^{(\text{cu})} = \frac{1}{N} \sum_{i=1}^{N} \|\nabla_{W_l} \mathcal{L}_{i,\text{OG}}\|. \tag{8}$$

We combine the two normalized measures using a convex combination to balance update magnitude and representation sensitivity, resulting in a sensitivity score that is empirically more stable than relying on a single signal:

$$S_l^{(\text{cu})} = \alpha \cdot \frac{\text{Diff}_l^{(\text{cu})}}{\max_l \text{Diff}_l^{(\text{cu})}} + (1-\alpha) \cdot \frac{\text{GradNorm}_l^{(\text{cu})}}{\max_l \text{GradNorm}_l^{(\text{cu})}}. \tag{9}$$

In addition, we utilize an exponential moving average (EMA) to update the critical-layer score as follows:

$$\text{PC}_l \leftarrow \beta \cdot \text{PC}_l + (1-\beta) \cdot S_l^{(\text{cu})}. \tag{10}$$

The server performs this procedure every R rounds to periodically update the layer relevance scores. After training, the top-K layers with the highest scores are selected as critical layers. Moreover, given the classifier's sensitivity to heterogeneous data distributions, we always include it as a critical layer. The selection of the critical layer set L_f is as follows:

$$L_f = w_{\{top-K(PC_l),-1\}}, \tag{11}$$

where $top - K(PC_l)$ is the index of top-K layer and -1 is the classifier.

3.2 Federated Unlearning via Sparse Critical-Layer Adapters

After PFUR identifies the critical layer set L_f, PFU-SKLA performs unlearning by freezing the global model and attaching sparse adapters to these layers. Although PFU-SKLA adopts an adapter-based mechanism similar to reversible methods such as FUSED, it differs in that the adapted layers are proactively identified during training based on representation-level sensitivity rather than selected post hoc from global parameter variations.

Specifically, for each critical layer $l \in L_f$, the server initializes an adapter A_l whose shape matches that of the original global layer weights w_l. All global parameters $\{w_l\}$ are frozen and remain unchanged during unlearning, while only the adapter parameters are trainable. We adopt element-wise sparsity for adapters. For each $l \in L_f$, we associate a binary mask $M_l \in \{0,1\}^{\text{shape}(w_l)}$ with $P(M_{l,j} = 1) = p$. Only the entries with $M_{l,j} = 1$ are allowed to be updated; the remaining entries are fixed to zero throughout unlearning. (Equivalently, one may parameterize $A_l = M_l \odot \tilde{A}_l$ and optimize $\tilde{A}_l$.)

During the unlearning stage, the server distributes the current adapters $\{A_l^t\}_{l\in L_f}$ (and masks $\{M_l\}_{l\in L_f}$, if needed) to the participating clients in the retained set $\mathcal{C}_{\text{remain}}$. Each client i reconstructs the effective model using the frozen base weights $\{w_l\}$ and updates only the adapter parameters by minimizing the loss on its retained data $D_{i,\text{remain}}$:

$$\mathcal{L}_{\text{unlearn}} = \mathcal{L}_{\text{CE}} + \lambda \cdot \mathcal{L}_{\text{OG}}, \tag{12}$$

where $\mathcal{L}_{\text{CE}}$ preserves performance on the retained data, and $\mathcal{L}_{\text{OG}}$ is reused to maintain the representation geometry established during training, stabilizing adapter updates rather than introducing additional regularization. Client i performs a gradient descent step to update the sparse adapter parameters:

$$A_{i,l}^{t+1} = A_{i,l}^t - \eta_t\big(M_l \odot \nabla_{A_{i,l}^t}\mathcal{L}_{\text{unlearn}}\big), \quad l \in L_f, \tag{13}$$

where $\odot$ denotes element-wise multiplication.

After local updates, the server aggregates the adapters from participating clients using FedAvg:

$$A_l^{t+1} = \sum_{i\in\mathcal{C}_{\text{remain}}} \frac{n_i}{N_{\text{remain}}} A_{i,l}^{t+1}, \quad l \in L_f, \tag{14}$$

where n_i is the retained data size of client i and $N_{\text{remain}} = \sum_{i\in\mathcal{C}_{\text{remain}}} n_i$.

For client unlearning, only clients in $\mathcal{C}_{\text{remain}}$ participate and each client trains on its retained data. For class unlearning or sample unlearning, the procedure is identical except that the target classes or samples to be forgotten are excluded from $D_{i,\text{remain}}$, and only the remaining data are used for local updates.

After T_u unlearning rounds, the final model is constructed as:

$$w_l^{\text{final}} = \begin{cases} w_l + A_l^{T_u}, & l \in L_f, \\ w_l, & l \notin L_f. \end{cases} \tag{15}$$

The forgetting effect is achieved solely through the learned sparse adapters $\{A_l^{T_u}\}$. As the global base parameters remain unchanged, PFU-SKLA is reversible:

$$w_{\text{unlearn},l} = w_l + A_l^{T_u}, \quad w_{\text{origin},l} = w_l. \tag{16}$$

The original model can be recovered by removing the adapters, enabling controllable and lightweight FU.

3.3 PFU-SKLA Overview

PFU-SKLA integrates PFUR and FU-SKLA into a unified unlearning pipeline. PFUR proactively prepares the model by decoupling representations and identifying key layers, while FU-SKLA updates only these layers using sparse adapters during unlearning. This yields efficient, localized, and reversible unlearning with minimal communication and computation. The procedure is summarized in Algorithm 1.

Algorithm 1: PFU-SKLA Algorithm

Input: Initial model w, training rounds T, unlearning rounds T_u, client set $\mathcal{C}$, remain-set clients $\mathcal{C}_{\text{remain}}$, hyperparameters $\beta, R, \lambda, \alpha$, sparsity rate p
Output: Unlearned model w^{final}

Stage 1: PFUR;
Initialize PC_l and L_f;
for $t = 1$ **to** T **do**
  for $i \in \mathcal{C}_t$ **do**
    perform local ORD training using Eq. 6 and upload w_i^{t+1};
  Update w^{t+1} via FedAvg;
  if $t \bmod R = 0$ **then**
    Update PC_l and L_f using Eq. 10 and 11;
Stage 2: FU-SKLA;
Freeze the global model w;
Initialize masks M_l with $P(M_{l,j} = 1) = p$ and adapters $A_l = 0$ for $l \in L_f$;
for $t = 1$ **to** T_u **do**
  for $i \in \mathcal{C}_{remain}$ **do**
    update $A_{i,l}$ using Eq. 13 and upload $\{A_{i,l}\}_{l\in L_f}$;
  Server aggregates $A_l \leftarrow \text{FedAvg}(\{A_{i,l}\})$ for $l \in L_f$;
Construct w^{final} using Eq. 15;
return w^{final};

4 Theoretical Analysis

This section provides a concise theoretical justification for the efficiency and effectiveness of PFU-SKLA. We first show that ORD in PFUR encourages decorrelated functional responses, and then explain why FU-SKLA yields low interference and exact reversibility. In essence, ORD reduces the overlap between gradient subspaces of retained and target knowledge, enabling sparse adapters to remove target knowledge with limited collateral damage.

4.1 Foundations of Proactive Unlearning Readiness

We adopt a functional mapping view for PFUR. Let $W_l \in \mathbb{R}^{C_{\text{out}}\times D}$ denote the flattened kernels of layer l, where each row $w_{l,i}$ is a filter vector and $C_{\text{out}} \le D$. Each filter induces a linear functional $F_{l,i}(x) = \langle w_{l,i}, x\rangle$ acting on a normalized local input $x \in \mathbb{R}^D$. ORD applies the orthogonality-guided regularizer $\mathcal{L}_{OG} = \|W_l W_l^\top - I\|_F^2$.

Lemma 1 (Approximate Orthonormality of Filters)**.** *If $\mathcal{L}_{OG} \to 0$, then for any filters $w_{l,i}, w_{l,j}$, we have $\langle w_{l,i}, w_{l,j}\rangle \approx \delta_{ij}$.*

Theorem 1 (Functional Decorrelation under Near-Isotropic Inputs)**.** *Assume the normalized local input has covariance Σ satisfying $\|\Sigma - I\|_2 \le \delta$. Given Lemma 1, for all $i \ne j$,*

$$\big|\mathbb{E}[F_{l,i}(x)F_{l,j}(x)]\big| \le |w_{l,i}^\top w_{l,j}| + \delta\,\|w_{l,i}\|_2\|w_{l,j}\|_2.$$

In particular, when $\mathcal{L}_{OG}$ is small (so $|w_{l,i}^\top w_{l,j}|$ is small), the cross-correlation is bounded by a small quantity.

Theorem 1 shows that ORD reduces cross-coupling among functional responses. This alone does not guarantee clean separation between target and retained knowledge, but offers a useful mechanism: if target- and retained-related information concentrates on different feature subsets, weaker cross-correlation reduces interference between them and can facilitate subsequent localization and selective editing.

Remark (Reduced Gradient Interference). By weakening coupling between target- and retained-dominated components, ORD makes the gradients induced by the retained objective T_1 and unlearning objective T_2 less aligned in expectation. This reduces overlap between their effective update subspaces and favors sparse, localized unlearning updates.

4.2 Stability and Reversibility of FU-SKLA

FU-SKLA freezes base parameters $\Theta^{(0)}$ and updates only sparse critical-layer adapters a injected via a fixed operator P, i.e., $\Theta = \Theta^{(0)} + Pa$. Under smoothness and local first-order approximation, and with an element-wise adapter mask activated at rate p, the expected retained-loss change induced by optimizing the unlearning objective is

$$\mathbb{E}[\Delta\mathcal{L}_{T_1}] \approx -\eta\, p \cdot \nabla_\Theta \mathcal{L}_{T_1}(\Theta^{(0)})^\top P \nabla_a \mathcal{L}_{T_2}(\Theta^{(0)}), \quad (17)$$

where η is the learning rate.

Eq. (17) captures the leading-order interference term for a single masked adapter step. Under standard smoothness and small-step conditions, the same cross-term accumulates along the unlearning trajectory, while higher-order terms are of order $\mathcal{O}(\eta^2)$.

Controlling Higher-Order Collateral Damage. The retained-loss change also includes a higher-order remainder $R(\Delta\Theta)$. With $\Delta\Theta = P\,\Delta a$ (and sparsity), smoothness yields $|R(\Delta\Theta)| \le \frac{L}{2}\|\Delta\Theta\|_2^2$. ORD encourages more concentrated, target-oriented updates, which reduces the magnitude of update components affecting retained knowledge and tightens this bound in expectation. Finally, since unlearning is implemented only through additive adapters while $\Theta^{(0)}$ remains unchanged, the original model is exactly restored by removing the adapters, yielding $\Theta_{\text{restored}} = \Theta^{(0)}$.

5 Experiments

5.1 Experimental Setting

In this section, we briefly summarize our experimental evaluation on FashionMNIST and CIFAR-10/100 under federated learning with 20 clients and full participation. We compare PFU-SKLA against representative FU baselines and report results from effectiveness, privacy, and efficiency perspectives.

5.2 Experimental Analysis

Effectiveness in Knowledge Retention and Forgetting

As shown in Table 1, PFU-SKLA achieves strong retention–forgetting performance across client, class, and sample unlearning. It attains high RA with near-zero FA on FashionMNIST, competitive RA on CIFAR-10 with effective forgetting (FA=4.15 for client and FA=0.00 for class), and maintains stable retention on the more challenging CIFAR-100 while keeping FA low (e.g., FA=0.15 for client unlearning). For sample unlearning, PFU-SKLA consistently yields strong retained-set performance, achieving 0A/PS of 100.00/100.00 on FashionMNIST, 71.66/76.47 on CIFAR-10, and competitive results on CIFAR-100 (e.g., 0A=75.00 on ResNet-18 and 0A/PS=100.00/70.00 on SimpleViT).

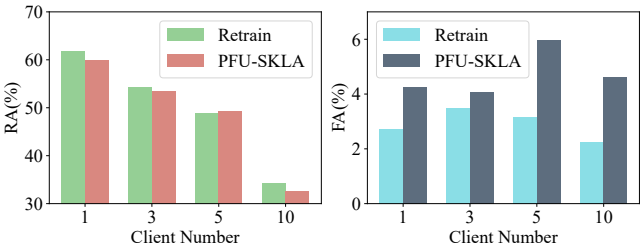


Figure 3: Impact of the number of forgotten clients on RA and FA for Retrain and PFU-SKLA on the CIFAR-10 with IID setting.

Computational and Communication Efficiency

PFU-SKLA consistently achieves the lowest computation and communication costs by updating only sparse critical-layer adapters. On CIFAR-10 and CIFAR-100, it substantially reduces Comp and communication compared with FedOSD and FUSED, and it also achieves very low communication overhead on FashionMNIST (Table 1).

Privacy Preservation and Relearning Accuracy

PFU-SKLA shows favorable privacy behavior, with relatively low MIA in client unlearning and low relearning accuracy in challenging settings, suggesting reduced membership leakage and limited recoverability of forgotten knowledge (Table 1).

Evaluating Knowledge Isolation and Retention

To study unlearning effects on unique and overlapping knowledge, we evaluate CIFAR-10 client unlearning in Table 2. The forgotten client removes all class-1 and 90% of class-0, while the remaining clients keep classes 2–9 and the remaining 10% of class-0. All methods achieve UF-Acc=0.00. PFU-SKLA yields the best O-Acc (19.81) and UR-Acc (60.48), indicating better retention of both overlapping (class-0) and unique retained knowledge (classes 2–9).

Performance of Multi-Client Unlearning

As shown in Figure 3, when the number of forgotten clients increases from 1 to 10 (50% of total clients), both Retrain and PFU-SKLA show a gradual decline in RA and a slight increase in FA. Specifically, PFU-SKLA maintains comparable RA to Retrain across all settings while achieving effective forgetting with only minor FA differences. Although Retrain attains marginally lower FA, it comes with much higher computational and communication overhead. Overall, the results demonstrate that PFU-SKLA achieves a balanced trade-off between retention and forgetting, maintaining stable performance as the number of forgotten clients grows while significantly improving unlearning efficiency.

Ablation Study

The ablation results in Table 3 support the contribution of ORD and our sensitivity-based key-layer identification (DSKLI). Removing ORD (w/o ORD) consistently reduces retention and privacy-related metrics across datasets, suggesting that ORD prepares more disentangled representations for stable unlearning. We also compare key-layer selection strategies during unlearning. Relative to sensitivity-based selection, the K random layers strategy leads to clear performance drops, especially for sample unlearning, where both

	Client Unlearning					Class Unlearning			Sample Unlearning			
	Retrain	FedEraser	FUSED	FedOSD	PFU-SKLA	Retrain	FUSED	PFU-SKLA		Retrain	FUSED	PFU-SKLA
FashionMNIST based LeNet												
RA(↑)	**99.52**	99.31	99.24	98.33	99.26	99.21	99.02	**99.63**	0A(↑)	100.00	100.00	**100.00**
FA (↓)	0.00	0.00	0.00	0.24	**0.00**	0.00	0.00	**0.00**	PS(↑)	93.68	93.68	**100.00**
MIA(↓)	55.12	**47.97**	51.35	50.97	50.27	99.60	96.99	**90.00**	MIA	99.21	99.79	**98.88**
ReA(↓)	97.93	99.18	99.78	97.69	**96.99**	100.00	100.00	**100.00**	ReA	**20.22**	80.21	43.60
Comp (↓)	882.12	1587.95	234.64	220.73	**186.45**	726.44	564.57	**151.53**	Comp	1476.22	1236.87	**375.04**
Comm(↓)	178.10K	178.10K	4.08K	178.10K	**3.58K**	178.10K	4.06K	**3.60K**	Comm	178.10K	4.07K	**3.59K**
CIFAR-10 based ResNet18												
RA(↑)	**61.78**	54.29	54.58	52.53	57.72	58.28	52.35	**59.40**	0A(↑)	53.33	58.14	**71.66**
FA (↓)	**2.73**	5.23	6.28	5.57	4.15	0.00	0.00	**0.00**	PS (↑)	67.75	64.62	**76.47**
MIA(↓)	74.52	56.43	67.37	**46.11**	51.19	**89.82**	97.85	89.97	MIA	99.16	97.48	**97.36**
ReA (↓)	47.23	40.61	**32.56**	42.11	46.89	100.00	100.00	**100.00**	ReA	100.00	100.00	**100.00**
Comp (↓)	9784.30	1749.44	1787.99	811.43	**231.77**	5453.17	1020.24	**481.83**	Comp	5510.64	2170.50	**478.22**
Comm (↓)	43.80M	43.80M	1.02M	43.80M	**0.021M**	43.80M	1.01M	**0.022M**	Comm	43.80M	1.02M	**0.021M**
CIFAR-100 based ResNet18												
RA (↑)	**32.48**	16.57	26.08	23.19	26.87	**24.75**	21.97	28.75	0A (↑)	50.00	**75.00**	**75.00**
FA (↓)	0.42	0.54	0.63	0.97	**0.15**	0.00	0.00	**0.00**	PS (↑)	62.00	**80.00**	75.00
MIA (↓)	76.81	47.83	58.65	**44.40**	68.61	85.68	**83.62**	94.40	MIA	98.46	98.56	**89.63**
ReA (↓)	15.14	14.13	13.32	**8.55**	12.67	100.00	100.00	**100.00**	ReA	3.09	**1.90**	3.01
Comp(↓)	13006.63	1733.84	1739.35	787.55	**192.82**	2583.95	933.39	**346.68**	Comp	4421.97	2931.76	**732.65**
Comm(↓)	43.95M	43.95M	1.02M	43.95M	**0.205M**	43.95M	1.01M	**0.208M**	Comm	43.95M	1.02M	**0.206M**
CIFAR-100 based SimpleViT												
RA(↑)	31.06	26.06	26.23	25.73	**27.18**	27.49	20.94	**30.25**	0A(↑)	50.00	50.00	**100.00**
FA (↓)	0.78	0.78	1.28	0.42	**0.36**	0.00	0.00	**0.00**	PS(↑)	50.00	50.00	**70.00**
MIA(↓)	66.90	66.9	65.56	**52.65**	64.14	**87.08**	87.70	92.31	MIA (↓)	99.06	98.93	**98.33**
ReA(↓)	18.29	16.29	18.64	**9.94**	14.34	100.00	100.00	**100.00**	ReA	2.85	**1.90**	2.61
Comp(↓)	5653.37	5653.37	833.50	779.77	**185.38**	2461.29	1075.32	**756.03**	Comp	4436.85	2912.11	**714.74**
Comm(↓)	43.95M	43.95M	1.03M	43.95M	**0.207M**	43.95M	1.02M	**0.209M**	Comm	43.95M	1.03M	**0.208M**

Table 1: Performance comparison of PFU-SKLA with baseline methods on client, class, and sample unlearning tasks under the IID setting. 0A denotes the accuracy on class 0 within S_r, and PS denotes the precision for predicting class 0 on S_r. ↑ indicates higher is better, ↓ indicates lower is better, **bold** denotes the top-1 result, and underline denotes the top-2 result.

Metric	Retrain	FedEraser	FUSED	FedOSD	PFU-SKLA
UF-Acc(↓)	0.00	0.00	0.00	0.00	0.00
O-Acc(↑)	15.94	12.39	18.64	11.23	**19.81**
UR-Acc(↑)	58.56	36.97	54.89	56.17	**60.48**

Table 2: Performance after client unlearning on CIFAR-10. UF-Acc: unique-forgotten (class-1); O-Acc: overlapping (class-0 on retained clients); UR-Acc: unique-retained (classes 2–9).

0A and PS decrease substantially on CIFAR-10/100 and SimpleViT, indicating stronger interference with retained knowledge. The Last-K layers strategy improves over random selection in some cases but remains consistently below PFU-SKLA, suggesting that a fixed heuristic may not reliably locate layers most responsible for target knowledge. Overall, DS-KLI provides a more consistent key-layer choice than random or last-layer heuristics under the same adapter budget, leading to improved retention–forgetting trade-offs.

6 Conclusion

In this work, we propose PFU-SKLA, a proactive, efficient FU framework. It introduces proactive unlearning readiness during training via ORD-based representation decoupling and sensitivity-based key-layer identification, enabling targeted, lightweight deployment-time unlearning. It supports exact reversibility because unlearning uses additive sparse adapters while the base model remains frozen, so removing them restores the pre-unlearning model. Experiments on FashionMNIST and CIFAR-10/100 show PFU-SKLA achieves competitive retention–forgetting trade-offs while reducing computation and communication versus representative baselines.

	Client Unlearning		Class Unlearning		Sample Unlearning	
	RA(↑)	FA(↓)	RA(↑)	FA(↓)	0A(↑)	PS(↑)
FashionMNIST based LeNet						
w/o ORD	98.30	0.00	98.94	0.00	98.22	96.49
Diff only	98.78	0.00	99.25	0.00	99.11	98.24
GradNorm only	98.95	0.00	99.38	0.00	99.45	98.81
K random layers	98.15	0.00	93.72	0.00	95.31	98.00
last-K layers	98.35	0.00	94.13	0.00	95.49	98.96
PFU-SKLA	99.26	0.00	99.63	0.00	100.00	100.00
CIFAR-10 based ResNet18						
w/o ORD	56.01	5.21	57.90	0.00	59.31	69.79
Diff only	56.86	4.68	58.65	0.00	65.48	73.13
GradNorm only	57.21	4.43	59.15	0.00	68.53	75.12
K random layers	57.50	4.24	59.10	0.00	57.70	53.00
last-K layers	57.51	4.21	59.23	0.00	63.45	60.22
PFU-SKLA	57.72	4.15	59.40	0.00	71.66	76.47
CIFAR-100 based ResNet18						
w/o ORD	26.02	0.70	27.07	0.00	40.00	50.00
Diff only	26.44	0.42	27.91	0.00	57.50	62.50
GradNorm only	26.65	0.28	28.32	0.00	66.25	68.75
K random layers	25.00	0.33	26.14	0.03	25.00	33.33
last-K layers	26.13	0.24	27.15	0.01	50.00	55.00
PFU-SKLA	26.87	0.15	28.75	0.00	75.00	75.00
CIFAR-100 based SimpleViT						
w/o ORD	26.92	0.91	27.46	0.00	70.00	70.00
Diff only	27.05	0.63	28.85	0.00	85.00	70.00
GradNorm only	27.11	0.49	29.55	0.00	92.50	70.00
K random layers	26.92	0.61	29.55	0.00	20.00	50.00
last-K layers	26.79	0.58	29.87	0.00	75.00	70.00
PFU-SKLA	27.18	0.36	30.25	0.00	100.00	70.00

Table 3: Ablation study on ORD and key-layer selection in PFU-SKLA. Diff only and GradNorm only use the individual criteria in Eqs. 7 and 8; parentheses report changes relative to PFU-SKLA.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (62271128, U2333207); the Major Science and Technology Project of Sichuan Province (2025ZDZX0093); the Science and Technology Program of Sichuan Province (2022ZDZX0004, 2025ZHRG0006); the "Challenge-Based" Project of Sichuan Provincial Science and Technology Program (2023YFG0374); the Regional Innovation Cooperation Project of Sichuan Province (2025YFHZ0302); and the Key Research and Development Support Program of Chengdu (2025-YF08-00128-GX, 2025-YF12-00029-RC, 2026-YF11-00032-HZ).

References

[Bourtoule *et al.*, 2021] Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE symposium on security and privacy (SP)*, pages 141–159. IEEE, 2021.

[Cao and Yang, 2015] Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE symposium on security and privacy*, pages 463–480. IEEE, 2015.

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

[Chowdhury *et al.*, 2024] Somnath Basu Roy Chowdhury, Krzysztof Choromanski, Arijit Sehanobish, Avinava Dubey, and Snigdha Chaturvedi. Towards scalable exact machine unlearning using parameter-efficient fine-tuning. *arXiv preprint arXiv:2406.16257*, 2024.

[Deng *et al.*, 2024] Zhipeng Deng, Luyang Luo, and Hao Chen. Enable the right to be forgotten with federated client unlearning in medical imaging. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 240–250. Springer, 2024.

[Eisenhofer *et al.*, 2025] Thorsten Eisenhofer, Doreen Riepel, Varun Chandrasekaran, Esha Ghosh, Olga Ohrimenko, and Nicolas Papernot. Verifiable and provably secure machine unlearning. In *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pages 479–496. IEEE, 2025.

[Gao *et al.*, 2022] Xiangshan Gao, Xingjun Ma, Jingyi Wang, Youcheng Sun, Bo Li, Shouling Ji, Peng Cheng, and Jiming Chen. Verifi: Towards verifiable federated unlearning. *IEEE Transactions on Dependable and Secure Computing*, 21:5720–5736, 2022.

[Graves *et al.*, 2021] Laura Graves, Vineel Nagisetty, and Vijay Ganesh. Amnesiac machine learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11516–11524, 2021.

[Gu *et al.*, 2024] Hanlin Gu, Gongxi Zhu, Jie Zhang, Xinyuan Zhao, Yuxing Han, Lixin Fan, and Qiang Yang. Unlearning during learning: An efficient federated machine unlearning method. *arXiv preprint arXiv:2405.15474*, 2024.

[Halimi *et al.*, 2022] Anisa Halimi, Swanand Kadhe, Ambrish Rawat, and Nathalie Baracaldo. Federated unlearning: How to efficiently erase a client in fl? *arXiv preprint arXiv:2207.05521*, 2022.

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

[Khalil *et al.*, 2025] Yasser H Khalil, Leo Brunswic, Soufiane Lamghari, Xu Li, Mahdi Beitollahi, and Xi Chen. Not: Federated unlearning via weight negation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 25759–25769, 2025.

[Koundinya Gundavarapu *et al.*, 2024] Saaketh Koundinya Gundavarapu, Shreya Agarwal, Arushi Arora, and Chandana Thimmalapura Jagadeeshaiah. Machine unlearning in large language models. *arXiv e-prints*, pages arXiv–2405, 2024.

[Lee *et al.*, 2019] Jaehoon Lee, Lechao Xiao, Samuel S. Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. In *NeurIPS*, pages 8570–8581, 2019.

[Li and Sui, 2025] Meng Li and Haochen Sui. Causal recommendation via machine unlearning with a few unbiased data. In *AAAI 2025 Workshop on Artificial Intelligence with Causal Techniques*, volume 2, 2025.

[Li *et al.*, 2024] Guihong Li, Hsiang Hsu, Chun-Fu Chen, and Radu Marculescu. Fast-ntk: Parameter-efficient unlearning for large-scale models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 227–234, 2024.

[Li *et al.*, 2025] Na Li, Chunyi Zhou, Yansong Gao, Hui Chen, Zhi Zhang, Boyu Kuang, and Anmin Fu. Machine unlearning: Taxonomy, metrics, applications, challenges, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*, 2025.

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

[Liu *et al.*, 2022] Yi Liu, Lei Xu, Xingliang Yuan, Cong Wang, and Bo Li. The right to be forgotten in federated learning: An efficient realization with rapid retraining. In *IEEE INFOCOM 2022-IEEE conference on computer communications*, pages 1749–1758. IEEE, 2022.

[Liu *et al.*, 2024] 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 Comput. Surv.*, 57(1), October 2024.

[Liu *et al.*, 2025] Sijia Liu, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Yuguang Yao, Chris Yuhao Liu, Xiaojun Xu, Hang Li, et al. Rethinking machine unlearning for large language models. *Nature Machine Intelligence*, pages 1–14, 2025.

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

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

[Nguyen *et al.*, 2025] Thanh Tam Nguyen, Thanh Trung Huynh, Zhao Ren, Phi Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen. A survey of machine unlearning. *ACM Transactions on Intelligent Systems and Technology*, 16(5):1–46, 2025.

[Pan *et al.*, 2025] 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, pages 19804–19812, 2025.

[Ranasinghe *et al.*, 2021] Kanchana Ranasinghe, Muzammal Naseer, Munawar Hayat, Salman Khan, and Fahad Shahbaz Khan. Orthogonal projection loss. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12333–12343, October 2021.

[Regulation, 2018] Protection Regulation. General data protection regulation. *Intouch*, 25:1–5, 2018.

[Ren *et al.*, 2024] Chao Ren, Han Yu, Hongyi Peng, Xiaoli Tang, Anran Li, Yulan Gao, Alysa Ziying Tan, Bo Zhao, Xiaoxiao Li, Zengxiang Li, et al. Advances and open challenges in federated learning with foundation models. *CoRR*, 2024.

[Su and Li, 2023] Ningxin Su and Baochun Li. Asynchronous federated unlearning. In *IEEE INFOCOM 2023-IEEE conference on computer communications*, pages 1–10. IEEE, 2023.

[Tarun *et al.*, 2023] Ayush K Tarun, Vikram S Chundawat, Murari Mandal, and Mohan Kankanhalli. Fast yet effective machine unlearning. *IEEE Transactions on Neural Networks and Learning Systems*, 35(9):13046–13055, 2023.

[Warnecke *et al.*, 2021] Alexander Warnecke, Lukas Pirch, Christian Wressnegger, and Konrad Rieck. Machine unlearning of features and labels. *arXiv preprint arXiv:2108.11577*, 2021.

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

[Wu *et al.*, 2022b] Leijie Wu, Song Guo, Junxiao Wang, Zicong Hong, Jie Zhang, and Yaohong Ding. Federated unlearning: Guarantee the right of clients to forget. *IEEE Network*, 36(5):129–135, 2022.

[Zhang *et al.*, 2022] Peng-Fei Zhang, Guangdong Bai, Zi Huang, and Xin-Shun Xu. Machine unlearning for image retrieval: A generative scrubbing approach. In *Proceedings of the 30th ACM international conference on multimedia*, pages 237–245, 2022.

[Zhang *et al.*, 2023] Haibo Zhang, Toru Nakamura, Takamasa Isohara, and Kouichi Sakurai. A review on machine unlearning. *SN Computer Science*, 4(4):337, 2023.

[Zhao *et al.*, 2023] Yian Zhao, Pengfei Wang, Heng Qi, Jianguo Huang, Zongzheng Wei, and Qiang Zhang. Federated unlearning with momentum degradation. *IEEE Internet of Things Journal*, 11(5):8860–8870, 2023.

[Zhao *et al.*, 2025] Yang Zhao, Jiaxi Yang, Yiling Tao, Lixu Wang, Xiaoxiao Li, Dusit Niyato, and H Vincent Poor. Exploring federated unlearning: Review, comparison, and insights. *IEEE Network*, 2025.

[Zhong *et al.*, 2025] 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 Computer Vision and Pattern Recognition Conference*, pages 30661–30670, 2025.