FedSOUL: Federated Continual Unlearning via Spectral Orthogonality

Haoran Shi[1], **Hao Zhu**[2], **Junru Zhang**[3], **Lizhen Cui**[4], **Yali Jiang**[4], **Yifei Zhang**[5,*], **Xiaoli Tang**[1,*], **Han Yu**[1]
[1]CCDS, Nanyang Technological University;
[2]Data61♡ CSIRO; [3]Zhejiang University; [4]Shandong University;
[5]School of Computer Science, Northwestern Polytechnical University;
SHIH0020@e.ntu.edu.sg, yifeiacc@gmail.com, xiaoli.tang@ntu.edu.sg

Abstract

As privacy, copyright, and safety requirements evolve, federated learning systems face a growing imperative to accommodate sequential unlearning requests. However, existing federated unlearning methods are designed for single-shot settings and their direct sequential application may lead to uncontrolled parameter-space overlap, causing cross-request interference that undermines prior forgetting. To address this, we propose the Spectral Orthogonality for Federated Continual unlearning (FedSOUL), a framework that reduces direct overlap among sequential unlearning updates. The core idea is to structure model updates in a shared orthonormal spectral basis, where each client is assigned a dedicated subspace in the frequency domain. Under this design, each unlearning request updates only its associated subspace, while all others remain unchanged, ensuring that sequential updates do not overlap in the allocated spectral coefficient space and thereby reducing cross-request interference. Beyond interference control, FedSOUL also improves efficiency by operating on sparse spectral coefficients, using FFT-based transforms for efficient computation and communicating only sparse coefficient updates. From a theoretical perspective, we relate continual unlearning to geometric conditions on update directions and show that our spectral construction satisfies update orthogonality by design, while functional preservation depends on bounded projected gradient leakage. Extensive experiments demonstrate that FedSOUL maintains stable unlearning–utility trade-offs across up to ten sequential requests and consistently outperforms competitive federated baselines.

1 Introduction

Federated learning (FL) enables collaborative model training across distributed participants while preserving data privacy [1–3]. Despite its advantages, real-world deployments increasingly require the ability to remove specific knowledge from trained models due to evolving privacy, copyright, and safety constraints [4]. Federated unlearning [5] addresses this need by selectively eliminating the influence of designated data while maintaining utility on the remaining knowledge. Most existing approaches consider unlearning as a single-shot operation [5–7]. However, in practice, unlearning requests arrive sequentially across clients, naturally forming a continual unlearning process. For example, a deployed federated model may first be required to remove data from one participant due to revoked consent, and later process additional removal requests from other participants over time. This continual setting introduces a fundamental challenge beyond conventional single-shot unlearning. Each update must simultaneously remove the current target, preserve utility on retained

*Corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

data, and maintain the forgetting effects of all previous requests. In naive implementations, repeated updates inevitably overlap in the parameter space, leading to severe cross-request interference that progressively degrades both forgetting quality and retained performance.

Related Works:[2] Existing federated unlearning methods fall into two main categories. Retraining-based approaches [6, 8] provide strong forgetting guarantees by excluding the target data and retraining the model from scratch, but incur prohibitive computational and communication costs, especially under sequential requests. In contrast, parameter manipulation-based methods [9–11] improve efficiency via gradient ascent, distillation, or sparse adapter tuning. However, they are primarily designed for single-shot unlearning and lack mechanisms to control interactions between updates across rounds. As a result, when applied sequentially, updates from different requests can overlap in the parameter space, leading to cross-request interference that degrades both retained utility and previously established forgetting. For example, Fused [11] applies sparsified adapters to critical layers for efficient single-shot unlearning, but without explicit control over update interactions, such sparsity patterns may collide across rounds, causing unpredictable interference.

To address this issue, we propose `FedSOUL`, a unified framework for continual unlearning in federated learning. The core idea is to structure model updates in a shared orthonormal spectral basis and assign each client a dedicated subspace in the frequency domain. Under this design, each unlearning request updates only the subspace associated with the target client, while all other subspaces remain fixed. This structured separation ensures that updates from different clients do not overlap in the allocated spectral coefficient space, reducing direct cross-request interference and helping preserve both retained utility and previously established forgetting. Beyond interference control, `FedSOUL` also improves efficiency through a frequency-domain design. By operating on sparse spectral coefficients, the framework uses FFT-based transforms for efficient computation, allowing clients to communicate only compact coefficient updates and reducing communication overhead. We evaluate `FedSOUL` under extensive client-level continual unlearning scenarios. Results show that `FedSOUL` consistently maintains strong forgetting persistence and retained utility as requests accumulate. Our key contributions are:

i. We formalize the federated continual unlearning problem, highlighting critical failure of existing single-shot methods when applied sequentially: accumulated cross-request interference leads to the re-emergence of previously forgotten knowledge and progressive degradation of model utility.

ii. We propose `FedSOUL`, which structures model updates in a shared orthonormal spectral basis and assigns each client a dedicated subspace. By restricting each unlearning request to its corresponding subspace, `FedSOUL` explicitly enforces update orthogonality across clients, reducing direct spectral-support interference and enabling more stable and effective accumulation of sequential unlearning updates.

iii. Extensive experiments on continual unlearning benchmarks demonstrate that `FedSOUL` consistently outperforms strong federated and centralized baselines, while maintaining strong *forgetting persistence*, and *utility preservation*.

2 Preliminaries

Federated learning setup. In a FL system with C clients $\mathcal{N} = \{1, \dots, C\}$, each client $k \in \mathcal{N}$ holds a local dataset $D_k = \{(\mathbf{x}_i, \mathbf{y}_i)\}$ and the server maintains a fully trained global model $\mathcal{G}_{\boldsymbol{\theta}}$ with parameters $\boldsymbol{\theta} = \{\boldsymbol{W}^{(\ell)}\}_{\ell=1}^{L}$, $\boldsymbol{W}^{(\ell)} \in \mathbb{R}^{m_\ell \times n_\ell}$. The empirical loss of client k is

$$L_k(\boldsymbol{\theta}) = \frac{1}{|D_k|} \sum_{(\mathbf{x},\mathbf{y}) \in D_k} \ell(\mathcal{G}_{\boldsymbol{\theta}}(\mathbf{x}), \mathbf{y}) \,. \tag{1}$$

Federated continual unlearning. We consider a sequence of client-level unlearning requests $\mathcal{R} = (c_1, \dots, c_T)$, $c_t \in \mathcal{N}$. At stage t, the forgetting set is $D^{u,(t)} = D_{c_t}$ and the retained set is $D^{r,(t)} = \{D_k\}_{k \in \mathcal{N} \setminus \{c_1, \dots, c_t\}}$. Starting from $\mathcal{G}_{\boldsymbol{\theta}^{(0)}}$, the model is updated sequentially as $\mathcal{G}_{\boldsymbol{\theta}^{(t-1)}} \to \mathcal{G}_{\boldsymbol{\theta}^{(t)}}$. Each updated model must simultaneously satisfy: (i) *Historical Forgetting Persistence*: maintain forgetting of all previously removed clients $\{c_1, \dots, c_{t-1}\}$ and current target c_t; (ii) *Retained-client Utility*

[2] Detailed related work is provided in the appendix.

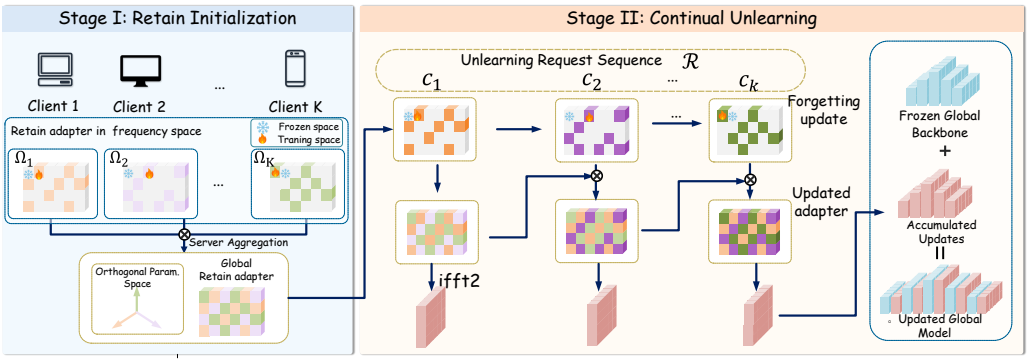


Figure 1: Federated continual unlearning framework. Left: the frequency domain is partitioned into disjoint spectral subspaces, each assigned to a client; within its subspace, each client learns a retain adapter, and the server aggregates them to form a global initialization. Right: for each sequential request, the target client optimizes a forgetting update within its assigned subspace; the server aggregates the spectral updates and applies ifft2 to obtain the stage-wise model update.

Preservation: preserve utility on $\mathcal{N} \setminus \{c_1, \ldots, c_t\}$. The stage-$t$ objective is

$$\min \sum_{k \in \mathcal{N} \setminus \{c_1, \ldots, c_t\}} L_k(\boldsymbol{\theta}^{(t)}), \qquad \max \; L_{c_t}(\boldsymbol{\theta}^{(t)}). \tag{2}$$

Challenges. The continual setting introduces two compounding difficulties absent from repeated applications of conventional single-shot unlearning. **(1) Cross-request interference**: without explicit structure, sequential updates overlap in parameter space, causing new unlearning steps to inadvertently restore previously forgotten knowledge and significantly degrade retained utility. **(2) Communication overhead**: re-engaging retained clients at each stage incurs $O(dCT)$ total communication cost, prohibitive in bandwidth-constrained federated networks.

3 Proposed Method: `FedSOUL`

We propose `FedSOUL`, a federated continual unlearning framework that reduces cross-request interference by assigning each client a dedicated, disjoint subspace in the 2D frequency domain. We begin by deriving the spectral parameterization directly from the optimizer update process, showing that the frequency-domain view emerges naturally from the structure of gradient-based updates.

3.1 From Gradient Updates to Spectral Coefficients

Consider a single weight layer $\boldsymbol{W}^{(\ell)} \in \mathbb{R}^{m_\ell \times n_\ell}$. A standard gradient-based update takes the form

$$\boldsymbol{W}^{(\ell)} \leftarrow \boldsymbol{W}^{(\ell)} + \Delta \boldsymbol{W}^{(\ell)}, \qquad \Delta \boldsymbol{W}^{(\ell)} = -\eta \, \boldsymbol{G}^{(\ell)}, \tag{3}$$

where $\boldsymbol{G}^{(\ell)} = \nabla_{\boldsymbol{W}^{(\ell)}} L$ is the gradient matrix and η is the step size. In this form, the update $\Delta \boldsymbol{W}^{(\ell)}$ is a dense $m_\ell \times n_\ell$ matrix, making communication of $\Delta \boldsymbol{W}^{(\ell)}$ to the server expensive and preventing principled control over which parts of the parameter space are modified.

Step 1: projecting the gradient into a complete basis. To gain explicit control over the update's structure, we express $\Delta \boldsymbol{W}^{(\ell)}$ in a fixed complete orthonormal basis. Let $\mathbf{F}_m \in \mathbb{C}^{m_\ell \times m_\ell}$ and $\mathbf{F}_n \in \mathbb{C}^{n_\ell \times n_\ell}$ be the 1D DFT matrices along rows and columns respectively, both unitary: $\mathbf{F}_m^H \mathbf{F}_m = \boldsymbol{I}_{m_\ell}$, $\mathbf{F}_n^H \mathbf{F}_n = \boldsymbol{I}_{n_\ell}$. Any matrix $\Delta \boldsymbol{W}^{(\ell)}$ can be expressed exactly as

$$\Delta \boldsymbol{W}^{(\ell)} = \mathbf{F}_m^H \underbrace{\mathbf{F}_m \, \Delta \boldsymbol{W}^{(\ell)} \, \mathbf{F}_n^H}_{=: \, \tilde{\Delta}^{(\ell)}} \mathbf{F}_n, \tag{4}$$

where $\tilde{\Delta}^{(\ell)} = \mathbf{F}_m \, \Delta \boldsymbol{W}^{(\ell)} \, \mathbf{F}_n^H = \texttt{fft2}(\Delta \boldsymbol{W}^{(\ell)}) \in \mathbb{C}^{m_\ell \times n_\ell}$ is the 2D DFT of the update. Since $\mathbf{F}_m, \mathbf{F}_n$ are unitary, this transformation is lossless: $\|\tilde{\Delta}^{(\ell)}\|_F = \|\Delta \boldsymbol{W}^{(\ell)}\|_F$ (Parseval's theorem).

Equation (4) shows that every weight update has an exact, equivalent representation as a set of complex spectral coefficients $\tilde{\Delta}^{(\ell)}$.

Step 2: sparsifying in the frequency domain. The full spectral representation $\tilde{\Delta}^{(\ell)}$ is still $m_\ell \times n_\ell$ and conveys no structural advantage. The key observation is that we can *approximate* the update by retaining only a sparse subset $\Omega^{(\ell)} \subseteq [m_\ell] \times [n_\ell]$ of spectral coefficients:

$$\boldsymbol{B}^{(\ell)} := P_{\Omega^{(\ell)}}\left(\tilde{\Delta}^{(\ell)}\right) = P_{\Omega^{(\ell)}}\left(\texttt{fft2}(\Delta \boldsymbol{W}^{(\ell)})\right), \tag{5}$$

P_Ω zeroes all entries outside Ω. The corresponding parameter-space update is then recovered as

$$\Delta \boldsymbol{W}^{(\ell)} \approx \mathbf{F}_m^H \, \boldsymbol{B}^{(\ell)} \, \mathbf{F}_n = \texttt{ifft2}\Big(\boldsymbol{B}^{(\ell)}\Big). \tag{6}$$

The approximation error equals the energy of $\tilde{\Delta}^{(\ell)}$ outside $\Omega^{(\ell)}$: $\|\Delta \boldsymbol{W}^{(\ell)} - \texttt{ifft2}(\boldsymbol{B}^{(\ell)})\|_F^2 = \sum_{(u,v)\notin\Omega^{(\ell)}} |\tilde{\Delta}^{(\ell)}[u,v]|^2$. When the selected support captures most of the gradient's spectral energy, the relative approximation error is small. To ensure real-valued updates, we impose Hermitian symmetry on $\boldsymbol{B}^{(\ell)}$: $\boldsymbol{B}^{(\ell)}[u,v] = \overline{\boldsymbol{B}^{(\ell)}[-u,-v]}$.

Step 3: the spectral update as a structured gradient step. Substituting Eq. (3) into Eq. 5, the sparse spectral coefficient at each active frequency (u,v) is computed as

$$\boldsymbol{B}^{(\ell)}[u,v] = -\eta \cdot \left[\texttt{fft2}\Big(\boldsymbol{G}^{(\ell)}\Big)\right]_{[u,v]}, \quad (u,v) \in \Omega^{(\ell)}. \tag{7}$$

This is precisely a gradient step performed *in the spectral domain*: apply the 2D FFT to the gradient, retain only the coefficients at $\Omega^{(\ell)}$, and apply the IFFT to obtain the parameter-space update. Critically, only $|\Omega^{(\ell)}| \ll m_\ell n_\ell$ scalars need to be computed, stored, and communicated — the IFFT reconstruction is performed at the server with no additional communication cost.

3.2 Disjoint Spectral Subspace Allocation

The derivation above shows that any gradient update can be represented as a set of spectral coefficients, and that restricting the update to a support $\Omega^{(\ell)}$ selects a specific subspace of the parameter space. The key structural property of this representation is: *Two updates whose spectral coefficients have disjoint supports are exactly Frobenius-orthogonal in parameter space.* This follows from the DFT isometry: for any A, B with $\mathrm{supp}(A) \cap \mathrm{supp}(B) = \emptyset$, $\langle \texttt{ifft2}(A), \texttt{ifft2}(B)\rangle_F = \langle A, B\rangle_F = 0$ (Lemma 4.1, proved in Appendix B). `FedSOUL` exploits this by pre-assigning each client a disjoint spectral support before any unlearning occurs. The server partitions the 2D frequency grid $[m_\ell] \times [n_\ell]$ into C disjoint subsets $\{\Omega_k^{(\ell)}\}_{k=1}^C$ of approximately equal size $|\Omega_k^{(\ell)}| \approx \lfloor m_\ell n_\ell / C \rfloor$, satisfying

$$\Omega_k^{(\ell)} \cap \Omega_{k'}^{(\ell)} = \emptyset, \quad \forall\, k \neq k',\ \forall\, \ell. \tag{8}$$

Each client k is restricted to spectral coefficients within $\Omega_k^{(\ell)}$ throughout both stages. Any update $\Delta \boldsymbol{W}_k^{(\ell)} = \texttt{ifft2}(\boldsymbol{B}_k^{(\ell)})$ with $\mathrm{supp}(\boldsymbol{B}_k^{(\ell)}) \subseteq \Omega_k^{(\ell)}$ therefore lies in a subspace exactly orthogonal to any other client's update subspace — no runtime projection or re-orthogonalization is required, and the guarantee holds for all stages simultaneously.

Stage I: Spectral Retention Initialization. Before processing unlearning requests, each client k learns a lightweight *retain adapter* by performing gradient descent restricted to its allocated support. Applying Eq. (7) in the *descent* direction, client k updates its spectral coefficients as

$$\boldsymbol{B}_k^{(\ell)}[u,v] \leftarrow \boldsymbol{B}_k^{(\ell)}[u,v] - \eta \left[\texttt{fft2}\Big(\boldsymbol{G}_k^{(\ell)}\Big)\right]_{[u,v]}, \quad (u,v) \in \Omega_k^{(\ell)}, \tag{9}$$

which minimizes the local loss L_k within the allocated subspace:

$$\min_{\{\boldsymbol{B}_k^{(\ell)}\}_\ell} L_k\Big(\boldsymbol{\theta} + \texttt{ifft2}(\boldsymbol{B}_k^{(\ell)})\Big), \quad \text{s.t. } \mathrm{supp}(\boldsymbol{B}_k^{(\ell)}) \subseteq \Omega_k^{(\ell)}. \tag{10}$$

After local optimization, client k uploads only the nonzero coefficients $\{\boldsymbol{B}_k^{(\ell)}\}_\ell$ to the server. The server sums the sparse spectral coefficients across all clients and recovers the global retain initialization via IFFT:

$$\Delta \boldsymbol{W}^{(\ell)} = \texttt{ifft2}\Big(\textstyle\sum_{k\in\mathcal{N}} \boldsymbol{B}_k^{(\ell)}\Big), \qquad \boldsymbol{\theta}^{(0)} \leftarrow \boldsymbol{\theta}^{(0)} + \Delta \boldsymbol{\theta}^{(0)}. \tag{11}$$

Disjointness of supports ensures the aggregation is exact: $\sum_k \boldsymbol{B}_k^{(\ell)}$ has support $\bigcup_k \Omega_k^{(\ell)}$ and the IFFT reconstruction recovers the sum of individual subspace updates without any mixing between clients.

Stage II: Continual Unlearning via Orthogonal Spectral Updates. `FedSOUL` processes the request sequence $\mathcal{R} = (c_1, \ldots, c_T)$ sequentially. At each stage t, client c_t performs a *forgetting update* by applying the same spectral gradient step Eq. (7) in the *ascent* direction, maximizing the loss on the forgotten data within its allocated support:

$$\boldsymbol{B}_{c_t}^{(\ell)}[u,v] \leftarrow \boldsymbol{B}_{c_t}^{(\ell)}[u,v] + \eta \left[\texttt{fft2}\Big(\boldsymbol{G}_{c_t}^{(\ell)}\Big)\right]_{[u,v]}, \quad (u,v) \in \Omega_{c_t}^{(\ell)}, \tag{12}$$

which solves

$$\max_{\{\boldsymbol{B}_{c_t}^{(\ell)}\}_\ell} L_{c_t}\Big(\boldsymbol{\theta}^{(t-1)} + \texttt{ifft2}(\boldsymbol{B}_{c_t}^{(\ell)})\Big), \quad \text{s.t. } \mathrm{supp}(\boldsymbol{B}_{c_t}^{(\ell)}) \subseteq \Omega_{c_t}^{(\ell)}. \tag{13}$$

The gradient ascent step in the spectral domain is computationally identical to the descent step Eq. (9) — only the sign changes. After optimization, client c_t uploads only $\{\boldsymbol{B}_{c_t}^{(\ell)}\}_\ell$ to the server. The server accumulates the update and reconstructs the new model:

$$\Delta\boldsymbol{\theta}^{(t)} = \Delta\boldsymbol{\theta}^{(t-1)} + \Delta\boldsymbol{\theta}_{c_t}, \qquad \boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}^{(0)} + \Delta\boldsymbol{\theta}^{(t)}. \tag{14}$$

Since $\mathrm{supp}(\boldsymbol{B}_{c_t}^{(\ell)}) \cap \mathrm{supp}(\boldsymbol{B}_{c_s}^{(\ell)}) = \emptyset$ for all $s \neq t$, the DFT isometry immediately gives $\langle \Delta\boldsymbol{W}_{c_t}^{(\ell)}, \Delta\boldsymbol{W}_{c_s}^{(\ell)} \rangle_F = 0$, so the cumulative update $\Delta\boldsymbol{\theta}^{(T)} = \sum_{t=1}^T \Delta\boldsymbol{\theta}_{c_t}$ is an orthogonal direct sum across all stages. This is the central structural guarantee: each unlearning update lives in a subspace perpendicular to all others, so zero direct overlap between updates on distinct supports, while their first-order functional effects are bounded by projected-gradient leakage (Theorem 4.2).

Communication efficiency. At each stage t, client c_t transmits only the $|\Omega_{c_t}^{(\ell)}|$ nonzero spectral coefficients per layer, reducing per-request communication from $O(m_\ell n_\ell)$ to $O(|\Omega_{c_t}^{(\ell)}|)$. Since the backbone $\boldsymbol{\theta}^{(0)}$ remains frozen and retained clients are never re-engaged during Stage II, the total communication over T requests scales as $O\Big(\sum_{t=1}^T |\Omega_{c_t}|\Big)$, compared to $O(dCT)$ for conventional sequential federated methods.

4 Theory Analysis

We analyze structural update orthogonality and functional preservation. Disjoint spectral allocation guarantees zero direct overlap between sequential updates (Lemma 4.1), while projected gradient leakage bounds the remaining first-order functional interference (Theorem 4.2). The analysis is further assessed empirically.

Lemma 4.1 (Spectral Orthogonality)**.** *For any two clients $k \neq k'$ with disjoint supports $\Omega_k^{(\ell)} \cap \Omega_{k'}^{(\ell)} = \emptyset$ for all ℓ,*

$$\langle \Delta\boldsymbol{\theta}_k,\ \Delta\boldsymbol{\theta}_{k'} \rangle = 0. \tag{15}$$

The proof applies the unitary DFT isometry (established in Section 3.1) to reduce the inner product of parameter-space updates to that of their spectral coefficients, where disjoint support immediately gives zero. The same isometry preserves gradient norms exactly. Thus, updates on pairwise disjoint supports form an orthogonal direct sum. However, this structural property does not ensure zero projected gradient leakage from retained or previously forgotten clients.

Let $g_k = \nabla L_k(\boldsymbol{\theta}^{(t-1)})$ and $\widetilde{g}_k = \mathcal{F}(g_k)$, where $\mathcal{F}$ applies the unitary spectral transform layerwise. Let Ω_{c_t} denote the collection of target-client supports across layers. At each unlearning stage, define

$$\rho_{q,t} := \frac{\|P_{\Omega_{c_t}} \widetilde{g}_q\|_F}{\|\widetilde{g}_q\|_F}, \qquad \gamma_t := \frac{\|P_{\Omega_{c_t}} \widetilde{g}_{c_t}\|_F}{\|\widetilde{g}_{c_t}\|_F}. \tag{16}$$

A ratio is set to zero when its denominator is zero. Here, $\rho_{q,t}$ measures the fraction of client q's gradient magnitude projected into the target support, while γ_t measures the fraction of the target-client gradient magnitude captured by that support. Neither quantity is assumed to be small.

For any nonzero update S_t supported on Ω_{c_t} in the spectral domain, let $\widetilde{S}_t = \mathcal{F}(S_t)$. By the Cauchy–Schwarz inequality,

$$\frac{\left|\left\langle P_{\Omega_{c_t}} \widetilde{g}_q, \widetilde{S}_t \right\rangle_F\right|}{\|\widetilde{g}_q\|_F \|\widetilde{S}_t\|_F} \leq \rho_{q,t}, \tag{17}$$

whenever $g_q \neq 0$. For complex spectral coefficients, inner products in the functional analysis below refer to the real part of the Hermitian inner product. Thus, $\rho_{q,t}$ bounds the normalized first-order interaction.

Theorem 4.2 (Projected-leakage bound)**.** *Suppose that the stage-t forgetting update is*

$$\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}^{(t-1)} + \eta \mathcal{F}^{-1}\left(P_{\Omega_{c_t}} \widetilde{g}_{c_t}\right), \tag{18}$$

where $\eta > 0$. If the client losses are locally smooth, then, for any retained or previously forgotten client q,

$$\left|L_q(\boldsymbol{\theta}^{(t)}) - L_q(\boldsymbol{\theta}^{(t-1)})\right| \leq \eta \rho_{q,t} \gamma_t \|g_q\|_F \|g_{c_t}\|_F + O(\eta^2). \tag{19}$$

For the target client,

$$L_{c_t}(\boldsymbol{\theta}^{(t)}) - L_{c_t}(\boldsymbol{\theta}^{(t-1)}) = \eta \gamma_t^2 \|g_{c_t}\|_F^2 + O(\eta^2). \tag{20}$$

Disjoint supports eliminate direct overlap between sequential updates, while $\rho_{q,t}$ and γ_t characterize the remaining first-order functional interaction. For fixed gradient norms and step size, small $\rho_{q,t}$ limits retained-client loss variation. When $\gamma_t > 0$, the update gives a positive first-order target-loss increase, which remains positive for sufficiently small η. These conclusions concern the specified one-step projected-ascent update.

Proof sketch. Let $\Delta\boldsymbol{\theta}_t = \eta \mathcal{F}^{-1}\left(P_{\Omega_{c_t}} \widetilde{g}_{c_t}\right)$. A first-order Taylor expansion gives

$$L_q(\boldsymbol{\theta}^{(t)}) - L_q(\boldsymbol{\theta}^{(t-1)}) = \langle g_q, \Delta\boldsymbol{\theta}_t \rangle_F + O(\eta^2).$$

Since $\mathcal{F}$ is unitary and $P_{\Omega_{c_t}}$ is an orthogonal projection,

$$\langle g_q, \Delta\boldsymbol{\theta}_t \rangle_F = \eta \left\langle P_{\Omega_{c_t}} \widetilde{g}_q, P_{\Omega_{c_t}} \widetilde{g}_{c_t} \right\rangle_F.$$

The Cauchy–Schwarz inequality yields

$$\left|\langle g_q, \Delta\boldsymbol{\theta}_t \rangle_F\right| \leq \eta \|P_{\Omega_{c_t}} \widetilde{g}_q\|_F \|P_{\Omega_{c_t}} \widetilde{g}_{c_t}\|_F.$$

Substituting the definitions of $\rho_{q,t}$ and γ_t, together with norm preservation, gives the retained-client bound. For $q = c_t$, $\langle g_{c_t}, \Delta\boldsymbol{\theta}_t \rangle_F = \eta \|P_{\Omega_{c_t}} \widetilde{g}_{c_t}\|_F^2 = \eta \gamma_t^2 \|g_{c_t}\|_F^2$, which gives the target-client result. □

Empirical projected leakage. To evaluate the realized leakage and target capture, we record $\rho_{q,t}$ and γ_t at each unlearning stage. We also examine the Spearman correlation between $\rho_{q,t}$ and the retained-client loss variation

$$\Delta L_{q,t} := \left|L_q(\boldsymbol{\theta}^{(t)}) - L_q(\boldsymbol{\theta}^{(t-1)})\right|.$$

Under the default FFT allocation, the mean and maximum leakage ratios are only 0.541% and 0.622%, respectively, while the mean and maximum target-capture ratios are 4.048% and 4.357%. Thus, the low retained-gradient leakage is not obtained by selecting an entirely inactive target support. In addition, leakage has a positive and statistically significant association with retained-client loss variation ($r_s = 0.213$, $p = 0.0159$). When the active support is enlarged to $|\Omega| = 5000$, both mean ρ and mean γ increase, and the association becomes stronger ($r_s = 0.409$, $p = 2.66 \times 10^{-5}$). These observations are consistent with the leakage-dependent bound in Theorem 4.2. This controlled comparison supports our default sparse allocation, which captures target-gradient information while limiting retained-client leakage. The correlations do not imply that leakage alone determines loss variation, since the bound also depends on gradient norms, target capture, and higher-order terms.

Random disjoint allocation. Random disjoint allocation provides a client-symmetric spectral budget without collecting client-specific gradient statistics or assuming separable dominant frequencies. Under uniform allocation of m out of d coefficients, every fixed gradient receives the same expected representational budget:

$$\mathbb{E}_{\Omega_k}\left[\|P_{\Omega_k} \widetilde{g}_k\|_F^2\right] = \frac{m}{d} \|\widetilde{g}_k\|_F^2. \tag{21}$$

This equality concerns expected captured energy; it does not imply zero functional leakage, which is quantified by $\rho_{q,t}$ and evaluated empirically above.

Table 1: Performance comparison on CIFAR-100 and TOFU under continual client unlearning. Lower S.U./D.U. indicates better forgetting; higher retained-utility metrics indicate better preservation. CIFAR-100 reports S.R./D.R., while TOFU reports R.D./R.A. **Bold** denotes the best result.

Dataset	Method	Forgotten Clients 1-5				Forgotten Clients 1-10				Forgotten Clients 1-15			
		unlearning effectiveness		utility preservation		unlearning effectiveness		utility preservation		unlearning effectiveness		utility preservation	
		S.U.↓	D.U.↓	S.R.↑	D.R.↑	S.U.↓	D.U.↓	S.R.↑	D.R.↑	S.U.↓	D.U.↓	S.R.↑	D.R.↑
CIFAR-100	FedAvg	45.7±0.9	20.7±0.0	51.7±0.5	27.9±0.1	45.2±0.1	20.1±4.2	56.7±6.4	28.8±0.0	51.0±2.5	27.2±1.8	60.9±3.9	27.1±6.5
	FedRetained	8.1±0.1	7.7±0.5	46.0±1.0	28.9±0.6	7.5±0.7	6.8±0.8	48.5±0.6	29.5±0.2	10.6±0.7	9.6±0.9	50.3±0.4	27.6±0.7
	PO	12.2±1.7	10.6±0.8	30.7±2.9	24.7±3.3	26.5±1.9	24.2±3.0	**45.2**±2.1	21.5±1.1	25.0±2.9	26.5±1.5	22.5±1.9	19.5±3.0
	NPO	16.4±2.3	16.6±0.6	32.5±1.2	23.0±4.9	12.5±1.1	16.0±3.2	28.8±3.0	24.7±0.7	25.6±1.7	22.2±2.1	35.6±2.5	28.7±1.0
	SOGD	**9.3**±2.2	**10.3**±2.5	36.7±0.5	26.6±1.4	25.4±1.4	21.6±0.2	41.9±1.8	25.2±2.4	23.4±2.9	**10.3**±0.7	23.5±1.7	18.4±2.7
	SOPO	11.3±0.8	11.9±0.5	31.9±2.0	24.3±1.7	12.4±0.5	12.7±0.2	27.4±1.2	22.4±2.3	18.9±2.4	16.5±2.8	25.9±1.4	24.4±0.6
	Exact-Fun	21.2±2.3	11.6±1.8	39.4±1.9	20.2±2.4	23.4±1.6	13.3±2.4	30.5±0.8	28.9±2.5	27.0±2.5	15.1±2.8	32.6±1.6	28.6±1.2
	Fused	12.5±2.2	13.8±1.2	39.6±8.1	27.3±1.6	18.4±4.3	14.3±0.6	37.2±3.2	17.8±0.5	19.1±0.3	17.1±1.3	26.3±2.5	29.0±0.6
	FedSOUL	10.5±0.6	**9.8**±1.2	**41.8**±1.3	**27.3**±2.8	**11.0**±1.2	**9.1**±2.6	40.3±1.5	**30.2**±2.0	**11.5**±1.3	11.3±3.7	**37.9**±2.2	**29.1**±0.3

		Forgotten Clients 1-2				Forgotten Clients 1-10				Forgotten Clients 1-20			
		S.U.↓	D.U.↓	R.D.↑	R.A.↑	S.U.↓	D.U.↓	R.D.↑	R.A.↑	S.U.↓	D.U.↓	R.D.↑	R.A.↑
TOFU	FedAvg	83.6±0.4	88.5±0.7	84.1±0.5	87.3±0.6	85.7±0.8	87.9±0.5	84.3±0.6	87.1±0.7	83.9±0.6	88.2±0.4	84.0±0.5	87.4±0.6
	FedRetained	61.2±0.3	59.6±0.8	79.4±0.9	70.2±0.4	53.4±0.6	52.3±0.4	78.5±0.8	78.9±0.5	57.0±0.5	51.3±0.8	78.7±0.6	71.6±0.2
	PO	18.7±1.4	19.8±2.1	73.1±1.0	76.4±1.8	65.9±6.7	64.8±7.1	79.3±1.6	75.2±3.4	61.6±3.2	58.9±2.6	74.8±1.4	79.1±1.9
	NPO	79.6±4.7	88.9±1.2	70.8±1.6	76.5±1.8	90.1±6.4	96.7±3.5	69.4±3.1	72.8±1.4	85.3±5.2	91.6±2.7	66.9±1.9	79.4±1.1
	SOGD	33.8±1.4	89.2±2.6	71.5±1.9	74.8±1.3	34.7±5.8	37.9±2.4	66.8±2.7	70.9±2.2	28.6±3.1	35.4±5.2	68.5±1.6	79.7±1.1
	SOPO	36.9±1.8	49.6±1.7	72.4±1.9	77.8±1.5	43.8±6.1	51.2±3.4	73.6±1.4	75.9±1.2	46.5±4.2	54.8±1.3	69.7±1.6	78.4±1.1
	FedEraser	22.1±1.4	20.4±1.2	74.6±1.5	80.1±1.3	25.8±1.2	30.1±2.8	71.8±1.4	77.5±1.2	27.2±1.1	35.6±1.7	70.5±1.3	81.4±1.1
	Exact-Fun	19.6±1.1	17.8±0.9	76.3±1.2	82.4±1.6	22.7±0.8	26.9±2.4	75.1±1.1	81.0±1.3	24.1±1.0	31.5±1.5	72.8±1.2	84.7±1.0
	Fused	24.8±1.6	22.9±1.4	70.6±1.7	76.9±1.2	27.3±1.1	31.6±3.1	69.8±1.5	75.4±1.0	28.5±1.3	37.9±1.8	67.1±1.4	79.3±1.1
	FedSOUL	**11.9**±1.6	**14.8**±1.2	**85.3**±1.3	**89.0**±0.1	**16.0**±2.2	**19.1**±1.6	**85.3**±0.5	**88.0**±0.9	**16.5**±0.3	**19.1**±0.7	**84.9**±0.2	**87.2**±0.4

Energy-aware allocation requires additional client-specific information and may create conflicts when multiple clients share dominant frequencies. It also does not necessarily reduce retained-gradient leakage, because target-dominant frequencies may contain substantial retained-gradient energy. We therefore retain random disjoint allocation for its simplicity, client symmetry, and exact update separation, while empirically evaluating functional leakage in the tested settings.

With C disjoint allocated supports, exact structural orthogonality holds for up to C sequential requests using distinct supports. Support reuse or overlap removes this guarantee (Appendix B.7).

5 Experiment

Datasets. We evaluate FedSOUL on CIFAR-100 [12] for visual unlearning and TOFU [13] for textual unlearning. We use ResNet-18 as the backbone for CIFAR-100, and follow prior work [13, 14] by adopting LLaMA2-7B [15] as the base model for TOFU.

Baselines. We compare FedSOUL against both centralized unlearning methods (PO [16], NPO [17], SOGD [14], and SOPO [14]) adapted to the federated setting and native federated unlearning baselines (FedEraser [18], Exact-Fun [19], and Fused [11]). We include FedAvg as the standard federated training backbone and FedRetained as an oracle upper bound trained solely on retained data, approximating a model never trained on the forgotten data.

Evaluation Metrics. We evaluate forgetting effectiveness using Sample-level Unlearning (**S.U.**) and Distribution-level Unlearning (**D.U.**) accuracy on the forget set, where lower values indicate more complete unlearning. For utility preservation, we use Sample-level Retaining (**S.R.**) and Distribution-level Retaining (**D.R.**) for CIFAR-100, and Retained Distribution (**R.D.**) and Real Authors (**R.A.**) for TOFU, where higher values indicate better retained utility. All experiments use 50 clients with non-IID data ($\alpha = 0.01$), averaged over three random seeds. Our code is available at https://github.com/final408/FedSOUL-Federated-Continual-Unlearning-via-Spectral-Orthogonality.git.

5.1 Main Results and Discussion

As shown in Table 1, FedSOUL consistently outperforms all baselines under continual unlearning across all settings, where "Forgotten Clients a–b" denotes the sequential unlearning of clients $\{c_a, \ldots, c_b\}$. On CIFAR-100, FedSOUL maintains a low D.U. as the forgotten set grows to clients 1–15, achieving 11.3 compared to 15.1 (Exact-Fun) and 17.1 (Fused), while preserving strong retained utility. The improvement is particularly notable on TOFU, where FedSOUL obtains the best

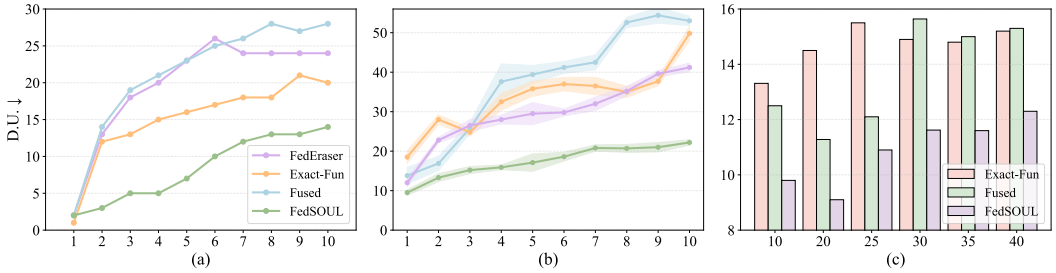


Figure 2: (a,b) D.U. of the first forgotten client over sequential unlearning requests (1–10) on CIFAR-100 and TOFU, respectively. (c) D.U. of the first 10 forgotten clients as the total number of forgotten clients increases on CIFAR-100.

Table 2: Effect of spectral support design in continual unlearning on CIFAR-100.

Method	D.U. ↓	D.R. ↑
Random sparse	20.9	17.4
Disjoint-50%	17.1	19.2
Disjoint-25%	16.2	20.0
`FedSOUL` (disjoint-0%)	**10.4**	**27.1**

Table 3: Performance across scale of the sparse coefficient matrix $\|\Omega_t\|$ on TOFU under Forgotten Clients 1-2.

n	S.U.↓	D.U.↓	R.D.↑	R.A.↑
20,000	20.6	18.3	78.7	86.0
50,000	14.9	17.9	82.8	86.8
70,000	13.4	14.6	84.1	88.7
100,000	**10.7**	**12.2**	**86.6**	**90.1**

forgetting performance with S.U./D.U. and the highest retained utility with R.D./R.A. under the largest forgotten-client group.

Federated baselines (FedEraser, Exact-Fun, Fused) preserve retained utility relatively well, but their forgetting performance remains weaker due to insufficient control over cross-request interference. In contrast, `FedSOUL` isolates sequential updates in structured spectral subspaces, achieving a better balance between forgetting persistence and utility preservation. This allows the model to remove accumulated forgotten clients while maintaining stable performance on retained clients.

5.2 Continual Unlearning Analysis

Figures 2(a) and 2(b) evaluate the forgetting persistence of the first forgotten client across sequential unlearning requests from stage 1 to stage 10. Compared with competing methods, `FedSOUL` consistently maintains lower D.U. values, indicating stronger forgetting persistence and reduced cross-request interference. Furthermore, as the total number of forgotten clients increases from 10 to 40 in Fig. 2(c), D.U. on the initially forgotten clients, i.e., clients 0–9, exhibits noticeable variation across all methods. This shows that different unlearning requests are assigned to structured spectral subspaces with limited overlap in `FedSOUL`, so later updates may introduce bounded perturbations to earlier forgetting effects, but are less likely to directly overwrite or restore the removed knowledge. This property is crucial for continual federated unlearning, as it allows the model to accommodate new requests while preserving the effectiveness of prior forgetting.

5.3 Ablation Study & Key Hyperparameter Sensitivity

Effect of Orthogonal Spectral Support. Table 2 compares four spectral support designs: *Random sparse* randomly selects frequency coefficients without structural constraints; *Disjoint-p%* allows a p% overlap between clients; and `FedSOUL` enforces full disjointness (p=0) by construction. The results demonstrate the necessity of disjoint spectral allocation under the 1–5 client unlearning setting. Without structured supports, the random sparse adapter baseline suffers from severe cross-request interference, yielding the worst D.U. and retained utility D.R.. As overlap ratio decreases from 50% to 0%, both metrics improve consistently, confirming that interference accumulates in proportion to support collision across sequential requests. `FedSOUL`, which enforces strict disjointness by construction, achieves the best result on both metrics simultaneously, demonstrating that orthogonal spectral support improves forgetting persistence and utility preservation in tandem rather than trading one off against the other.

Scalability with sparse coefficient matrix. Table 3 reports the impact of varying the sparse coefficient matrix scale $\|\Omega_t\|$ on TOFU under Forgotten Clients 1–2. As $\|\Omega_t\|$ increases, both S.U. and D.U. consistently decrease, while R.D. and R.A. steadily improve overall. Specifically, enlarging

$\|\Omega_t\|$ from 20,000 to 100,000 reduces S.U. from 20.6 to 10.7 and D.U. from 18.3 to 12.2, while improving R.D. from 78.7 to 86.6 and R.A. from 86.0 to 90.1. Although slight fluctuations appear at intermediate scales, the overall trend demonstrates that larger sparse spectral support enables more effective continual unlearning while better preserving retained knowledge and downstream utility. These results validate the scalability and effectiveness of the proposed frequency-conditioned adapter design.

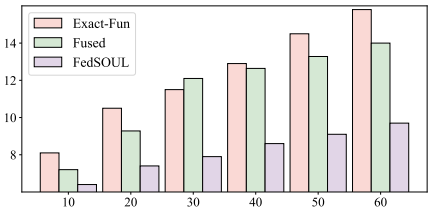


Figure 3: Comparison of D.U. (lower is better) on CIFAR-100 as the number of clients increases. The unlearning ratio is fixed at 40%, i.e., 40% of clients are sequentially selected as unlearning targets.

Scalability with Client Number. We evaluate the scalability of different methods on CIFAR-100 by increasing the number of clients while fixing the unlearning ratio to 40%. As shown in Fig. 3, all methods exhibit degraded D.U. performance as the number of clients increases. This is expected since each client holds fewer samples and the data distribution becomes more heterogeneous, making distributional unlearning more challenging. Despite this, `FedSOUL` consistently achieves the lowest D.U. across all settings, indicating more effective removal of target client distributions. In contrast, Exact-Fun and Fused suffer from steadily increasing D.U., suggesting that their unlearning performance deteriorates under larger federated systems. Notably, the gap between `FedSOUL` and baselines becomes more pronounced as the number of clients grows, demonstrating the superior scalability of our spectral orthogonality design in handling large-scale federated continual unlearning.

5.4 Communication Efficiency

As shown in Fig. 4, `FedSOUL` achieves the best D.U.–communication trade-off among all compared methods. Exact-Fun and FedEraser incur substantial communication overhead, as they require repeated rounds of client-server synchronization or the transmission of dense model updates throughout the entire unlearning process. Fused alleviates this burden by transmitting sparse unlearning adapters, yet its randomly selected frequency components risk overlapping across sequential unlearning requests, leading to severe cross-request interference that progressively degrades forgetting quality over time. In contrast, `FedSOUL` avoids full-model transmission entirely by restricting optimization and communication to compact spectral coefficients within each client's pre-allocated FourierFT subspace. This design not only significantly minimizes the volume of communicated parameters but also ensures that updates from different clients remain orthogonal, effectively eliminating interference without introducing additional coordination cost or extra synchronization rounds. As a result, `FedSOUL` achieves the lowest communication cost among all baselines while simultaneously maintaining a highly competitive forgetting score, demonstrating that the reduction in communication overhead does not come at the expense of unlearning effectiveness or retained model utility. These results further highlight `FedSOUL` as a highly communication-efficient and practically scalable solution well suited to the realistic demands of federated continual unlearning in resource-constrained environments.

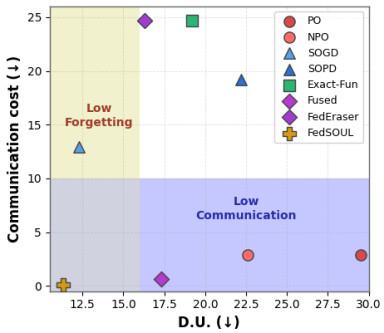


Figure 4: D.U.-communication trade-off on CIFAR-100.

6 Conclusion

This paper studies federated continual unlearning, where sequential requests induce cross-request interference that undermines both forgetting persistence and retained utility over time. To address this, we propose `FedSOUL`, a structured framework that parameterizes model updates in a shared spectral basis and assigns each client a dedicated subspace, enabling non-overlapping spectral updates and more stable accumulation across sequential unlearning stages. We further show that continual unlearning can be related to geometric conditions on update directions, and that our spectral construction satisfies update orthogonality by design while requiring bounded projected gradient

leakage for functional preservation. This establishes a structured and theoretically grounded approach to continual unlearning and provides a scalable foundation for federated systems.

Acknowledgements and Disclosures

Funding. The research is supported by the Ministry of Education, Singapore, under its Academic Research Fund Tier 1 (RG101/24). Yifei Zhang was supported by the National Natural Science Foundation of China (No. 62606400) and Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China (No. JYB2025XDXM202).

References

[1] Tao Fan, Hanlin Gu, Xuemei Cao, Chee Seng Chan, Qian Chen, Yiqiang Chen, Yihui Feng, Yang Gu, Jiaxiang Geng, Bing Luo, et al. Ten challenging problems in federated foundation models. *TKDE*, 37(7):4314–4337, 2025.

[2] Yifei Zhang, Dun Zeng, Jinglong Luo, Xinyu Fu, Guanzhong Chen, Zenglin Xu, and Irwin King. A survey of trustworthy federated learning: Issues, solutions, and challenges. *ACM Transactions on Intelligent Systems and Technology*, 15(6):1–47, 2024.

[3] Haoran Shi, Junru Zhang, Xiaoli Tang, Yifei Zhang, Hao Zhu, Yabo Dong, Lizhen Cui, and Han Yu. Federated domain generalization for time-series classification via dynamics-to-domain generation. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 4219–4228, 2026.

[4] Xudong Pan, Mi Zhang, Shouling Ji, and Min Yang. Privacy risks of general-purpose language models. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 1314–1331. IEEE, 2020.

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

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

[7] Yifei Zhang, Hao Zhu, Haoran Shi, Yanyu Chen, Xiaolin Han, Yupei Zhang, Lingyun Song, Wenxuan Wang, Chao Zhou, Xuequn Shang, et al. Flash: Fast generative retrieval via autoregressive semantic hashing with provably distance bounds. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 6594–6605, 2026.

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

[9] Junxiao Wang, Song Guo, Xin Xie, and Heng Qi. Federated unlearning via class-discriminative pruning. In *Proceedings of the ACM web conference 2022*, pages 622–632, 2022.

[10] Anisa Halimi, Swanand Kadhe, Ambrish Rawat, and Nathalie Baracaldo. Federated unlearning: How to efficiently erase a client in fl? In *Proceedings of the Thirty-ninth International Conference on Machine Learning (ICML'22)*, pages 1–10, 2022.

[11] 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.

[12] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[13] Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C Lipton, and J Zico Kolter. Tofu: A task of fictitious unlearning for llms. *arXiv preprint arXiv:2401.06121*, 2024.

[14] Jinghan Jia, Yihua Zhang, Yimeng Zhang, Jiancheng Liu, Bharat Runwal, James Diffenderfer, Bhavya Kailkhura, and Sijia Liu. Soul: Unlocking the power of second-order optimization for llm unlearning. *arXiv preprint arXiv:2404.18239*, 2024.

[15] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[16] Ronen Eldan and Mark Russinovich. Who's harry potter? approximate unlearning in llms. *arXiv preprint arXiv:2310.02238*, 2023.

[17] Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning. *arXiv preprint arXiv:2404.05868*, 2024.

[18] 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.

[19] Zuobin Xiong, Wei Li, Yingshu Li, and Zhipeng Cai. Exact-fun: An exact and efficient federated unlearning approach. In *2023 IEEE International Conference on Data Mining (ICDM)*, pages 1439–1444, 2023.

[20] Jia-Chen Gu, Hao-Xiang Xu, Jun-Yu Ma, Pan Lu, Zhen-Hua Ling, Kai-Wei Chang, and Nanyun Peng. Model editing can hurt general abilities of large language models. *arXiv preprint arXiv:2401.04700*, 2024.

[21] Akshat Gupta, Anurag Rao, and Gopala Anumanchipalli. Model editing at scale leads to gradual and catastrophic forgetting. *arXiv preprint arXiv:2401.07453*, 2024.

[22] Zhuang Qi, Ying-Peng Tang, Lei Meng, Guoqing Chao, Lei Wu, Han Yu, and Xiangxu Meng. From selection to scheduling: Federated geometry-aware correction makes exemplar replay work better under continual dynamic heterogeneity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17494–17504, June 2026.

[23] 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.

[24] General Data Protection Regulation GDPR. General data protection regulation. *URL: https://gdpr-info. eu/[accessed 2020-11-21]*, 2018.

[25] Stuart L Pardau. The california consumer privacy act: Towards a european-style privacy regime in the united states. *J. Tech. L. & Pol'y*, 23:68, 2018.

[26] Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345, 2024.

[27] Jiaao Chen and Diyi Yang. Unlearn what you want to forget: Efficient unlearning for llms. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

[28] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372, 2022.

[29] Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D Li, Ann-Kathrin Dombrowski, Shashwat Goel, Long Phan, et al. The wmdp benchmark: Measuring and reducing malicious use with unlearning. *arXiv preprint arXiv:2403.03218*, 2024.

[30] 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*, pages 9304–9312, 2020.

[31] Yuanshun Yao, Xiaojun Xu, and Yang Liu. Large language model unlearning. In *Socially Responsible Language Modelling Research*, 2023.

[32] Charles Yu, Sullam Jeoung, Anish Kasi, Pengfei Yu, and Heng Ji. Unlearning bias in language models by partitioning gradients. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6032–6048, 2023.

[33] Pratiksha Thaker, Yash Maurya, and Virginia Smith. Guardrail baselines for unlearning in llms. *arXiv preprint arXiv:2403.03329*, 2024.

[34] Martin Pawelczyk, Seth Neel, and Himabindu Lakkaraju. In-context unlearning: Language models as few shot unlearners. *ICML*, 2024.

[35] Sijia Liu, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Xiaojun Xu, Yuguang Yao, Hang Li, Kush R Varshney, et al. Rethinking machine unlearning for large language models. *arXiv preprint arXiv:2402.08787*, 2024.

[36] Lichao Sun, Yue Huang, Haoran Wang, Siyuan Wu, Qihui Zhang, Chujie Gao, Yixin Huang, Wenhan Lyu, Yixuan Zhang, Xiner Li, et al. Trustllm: Trustworthiness in large language models. *arXiv preprint arXiv:2401.05561*, 2024.

A Related Work

Continual Learning and Catastrophic Forgetting. The continual unlearning challenge is closely related to catastrophic forgetting in continual learning [20, 21]. However, while continual learning[22] seeks to preserve knowledge across tasks, continual unlearning requires the opposite: selectively forgetting specific knowledge while maintaining everything else. This difference makes continual unlearning even more challenging, as it combines the need for targeted degradation with strong utility preservation. Recent studies have begun to examine sequential unlearning scenarios [20], showing that naive reuse of single-task methods leads to cumulative degradation of utility across unaffected domains. These works, however, remain primarily empirical and lack theoretical grounding. They do not provide a principled framework for analyzing or mitigating interference between sequential unlearning operations, leaving open the question of how to design methods with provable guarantees in the continual setting.

Unlike continual learning, where stability and plasticity trade-offs have been formalized, continual unlearning has not yet been framed in terms of explicit conditions that algorithms must satisfy. This missing formulation is precisely what we establish in Section 2.

Federated Unlearning. Machine unlearning [23] addresses the "right to be forgotten" [24, 25] by enabling the removal of specific data influences from trained models. Early approaches relied on exact unlearning through retraining from scratch, which is computationally prohibitive for large-scale LLMs [26]. This motivated the development of approximate unlearning methods that aim to remove the effect of target data with significantly reduced overhead. Current LLM unlearning techniques can be broadly divided into two paradigms: **parameter optimization methods** and **in-context learning methods**. Parameter optimization approaches [27, 16, 14, 17, 28, 29] directly modify model weights through strategies such as gradient ascent on unlearning data [30, 31], preference optimization with shuffled or rejection labels [16, 17], or localizing parameters and updating only selected subsets [32]. In-context learning methods [33, 34], by contrast, modify prompts to elicit refusal responses for undesired content without changing the underlying parameters. While parameter optimization methods typically achieve stronger forgetting performance than in-context methods, they generally assume access to substantial retained datasets to preserve model utility. This assumption is increasingly unrealistic for LLMs trained on massive, proprietary corpora [35, 36]. Moreover, existing approaches largely treat unlearning as isolated, single-task operations and neglect the continual nature of real-world unlearning requests.

B Theory Analysis

B.1 Theoretical Conditions for Continual Unlearning

To make federated continual unlearning analytically tractable, we relax the stage-wise optimization objective into three measurable conditions covering the two requirements defined in Section 2.

Definition B.1 (Selective Forgetting)**.** *At stage t, the update is said to achieve selective forgetting for the current target client c_t if*

$$L_{c_t}(\boldsymbol{\theta}^{(t)}) \;\geq\; L_{c_t}(\boldsymbol{\theta}^{(t-1)}) + \epsilon_t, \qquad \epsilon_t > 0.$$

Definition B.2 (Utility Preservation)**.** *At stage t, the update is said to preserve utility if, for every retained client $k \in \mathcal{N} \setminus \{c_1, \ldots, c_t\}$,*

$$\left|L_k(\boldsymbol{\theta}^{(t)}) - L_k(\boldsymbol{\theta}^{(t-1)})\right| \;\leq\; \delta.$$

Definition B.3 (Historical Forgetting Persistence)**.** *At stage t, the update is said to preserve historical forgetting if, for every previously forgotten client c_s with $s < t$,*

$$L_{c_s}(\boldsymbol{\theta}^{(t)}) \;\geq\; L_{c_s}(\boldsymbol{\theta}^{(s)}) - \zeta.$$

Here, ϵ_t measures the forgetting margin of the current request, δ bounds the utility drift on retained clients, and ζ quantifies the extent to which previously forgotten knowledge may be unintentionally restored. This loss-increase condition is a first-order surrogate for forgetting, not a certified deletion guarantee. In practice, δ and ζ absorb higher-order approximation errors and stochasticity induced by local optimization.

B.2 Optimization View and First-Order Characterization

We now relate the above three conditions to the geometry of the stage-wise update. Let $\Delta\boldsymbol{\theta}_t = \boldsymbol{\theta}^{(t)} - \boldsymbol{\theta}^{(t-1)}$ denote the update introduced at stage t. For each client k, let $g_k(\boldsymbol{\theta}^{(t-1)}) = \nabla_{\boldsymbol{\theta}} L_k(\boldsymbol{\theta})\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(t-1)}}$ be the gradient of its empirical loss evaluated at the current model.

Under a first-order approximation around $\boldsymbol{\theta}^{(t-1)}$, we have

$$L_k(\boldsymbol{\theta}^{(t)}) \approx L_k(\boldsymbol{\theta}^{(t-1)}) + \langle g_k(\boldsymbol{\theta}^{(t-1)}), \Delta\boldsymbol{\theta}_t \rangle. \tag{22}$$

Eq. equation 22 shows that the effect of continual unlearning is determined by the alignment between the stage-wise update direction $\Delta\boldsymbol{\theta}_t$ and the client gradients. Based on this observation, the three continual unlearning requirements can be expressed as geometric conditions on $\Delta\boldsymbol{\theta}_t$.

Proposition B.4 (First-order characterization of continual unlearning)**.** *At stage t, suppose the update $\Delta\boldsymbol{\theta}_t$ satisfies*

$$\langle g_k(\boldsymbol{\theta}^{(t-1)}), \Delta\boldsymbol{\theta}_t \rangle = 0, \qquad \forall\, k \in \mathcal{N} \setminus \{c_1, \ldots, c_t\}, \tag{23}$$

and

$$\langle g_{c_s}(\boldsymbol{\theta}^{(t-1)}), \Delta\boldsymbol{\theta}_t \rangle = 0, \qquad \forall\, s < t. \tag{24}$$

Then, under the first-order approximation in Eq. (22), utility preservation and historical forgetting persistence hold exactly, i.e., $\delta = \zeta = 0$. Moreover, if

$$\langle g_{c_t}(\boldsymbol{\theta}^{(t-1)}), \Delta\boldsymbol{\theta}_t \rangle > 0, \tag{25}$$

then selective forgetting is achieved with margin

$$\epsilon_t \approx \langle g_{c_t}(\boldsymbol{\theta}^{(t-1)}), \Delta\boldsymbol{\theta}_t \rangle > 0.$$

Proof. For any retained client $k \in \mathcal{N} \setminus \{c_1, \ldots, c_t\}$, substituting Eq. (23) into Eq. (22) yields

$$L_k(\boldsymbol{\theta}^{(t)}) \approx L_k(\boldsymbol{\theta}^{(t-1)}),$$

which implies exact utility preservation under the first-order approximation.

Similarly, for any previously forgotten client c_s with $s < t$, Eq. (24) gives

$$L_{c_s}(\boldsymbol{\theta}^{(t)}) \approx L_{c_s}(\boldsymbol{\theta}^{(t-1)}).$$

Thus the current update does not decrease the loss of the previously forgotten client to first order. Applying this argument inductively over stages $s+1, \ldots, t$ gives

$$L_{c_s}(\boldsymbol{\theta}^{(t)}) \approx L_{c_s}(\boldsymbol{\theta}^{(s)}),$$

so historical forgetting is preserved with $\zeta = 0$ under the local first-order approximation.

Finally, for the current target client c_t, Eq. (25) implies

$$L_{c_t}(\boldsymbol{\theta}^{(t)}) \approx L_{c_t}(\boldsymbol{\theta}^{(t-1)}) + \langle g_{c_t}(\boldsymbol{\theta}^{(t-1)}), \Delta\boldsymbol{\theta}_t \rangle > L_{c_t}(\boldsymbol{\theta}^{(t-1)}),$$

which establishes selective forgetting with positive margin. This completes the proof. □

Proposition B.4 shows that continual unlearning can be achieved by constructing a stage-wise update that is aligned with the current target client's forgetting direction, while remaining orthogonal to the directions associated with retained clients and previously forgotten clients. We next show that such a construction can be realized explicitly in the spectral domain.

B.3 Spectral Parameterization

While the above analysis uses the vectorized model parameter $\boldsymbol{\theta}$, practical computation is performed layer-wise. Let the global model consist of L layers, and for each layer ℓ, let

$$\boldsymbol{W}^{(\ell)} \in \mathbb{R}^{m_\ell \times n_\ell}$$

denote the corresponding weight matrix. The vectorized parameter $\boldsymbol{\theta}$ is obtained by concatenating the vectorized layer-wise parameters:

$$\boldsymbol{\theta} = \left[\mathrm{vec}(\boldsymbol{W}^{(1)})^\top, \mathrm{vec}(\boldsymbol{W}^{(2)})^\top, \ldots, \mathrm{vec}(\boldsymbol{W}^{(L)})^\top\right]^\top.$$

Hence, the vectorized and matrix views are equivalent via the $\mathrm{vec}(\cdot)$ operator.

For each layer ℓ, we introduce square unitary basis matrices

$$\boldsymbol{U}_L^{(\ell)} \in \mathbb{C}^{m_\ell \times m_\ell}, \qquad \boldsymbol{U}_R^{(\ell)} \in \mathbb{C}^{n_\ell \times n_\ell},$$

where the practical implementation uses the Fourier bases described in Section 3.1. For complex matrices we use $\langle A, B\rangle_F := \mathrm{Re}\,\mathrm{tr}(A^*B)$, which coincides with the usual real Frobenius inner product after inverse FFT reconstruction. We parameterize the stage-wise update associated with request c_t as

$$\Delta \boldsymbol{W}_{c_t}^{(\ell)} = \boldsymbol{U}_L^{(\ell)} \boldsymbol{S}_{c_t}^{(\ell)} (\boldsymbol{U}_R^{(\ell)})^*, \qquad \boldsymbol{S}_{c_t}^{(\ell)} \in \mathbb{C}^{m_\ell \times n_\ell}. \tag{26}$$

Here, $\boldsymbol{S}_{c_t}^{(\ell)}$ is a sparse coefficient matrix in the spectral domain. Its nonzero entries are restricted to a support set

$$\Omega_{c_t}^{(\ell)} \subseteq [m_\ell] \times [n_\ell],$$

which is allocated exclusively to request c_t. We define the sparsity ratio as

$$\rho_\ell = \frac{|\Omega_{c_t}^{(\ell)}|}{m_\ell n_\ell},$$

and the total spectral budget as

$$\mathcal{K} = \sum_{\ell=1}^{L} m_\ell n_\ell.$$

Let $P_\Omega^{(\ell)}$ denote the projector onto entries in $\Omega^{(\ell)}$.

This parameterization decouples continual unlearning from dense full-model optimization: instead of updating all entries of $\boldsymbol{W}^{(\ell)}$, the method only optimizes sparse spectral coefficients inside the allocated support set.

B.4 Orthogonality by Construction

Proof of Lemma 4.1. For two clients $k \neq k'$, write their layer-wise spectral coefficients as $\boldsymbol{S}_k^{(\ell)}$ and $\boldsymbol{S}_{k'}^{(\ell)}$. By the unitary invariance of the Frobenius inner product,

$$\langle \Delta \boldsymbol{W}_k^{(\ell)}, \Delta \boldsymbol{W}_{k'}^{(\ell)} \rangle_F = \langle \boldsymbol{S}_k^{(\ell)}, \boldsymbol{S}_{k'}^{(\ell)} \rangle_F.$$

Since $\Omega_k^{(\ell)} \cap \Omega_{k'}^{(\ell)} = \emptyset$, the coefficient-wise product is zero at every frequency, hence the layer-wise inner product is zero. Summing over layers gives $\langle \Delta\boldsymbol{\theta}_k, \Delta\boldsymbol{\theta}_{k'} \rangle = 0$.

Let $\boldsymbol{G}_k^{(\ell)}$ denote the gradient of client k on layer ℓ. The Frobenius inner product between a client gradient and the stage-wise update in Eq. (26) can be factorized through the spectral basis as

$$\langle \boldsymbol{G}_k^{(\ell)}, \Delta \boldsymbol{W}_{c_t}^{(\ell)} \rangle_F = \left\langle (\boldsymbol{U}_L^{(\ell)})^* \boldsymbol{G}_k^{(\ell)} \boldsymbol{U}_R^{(\ell)}, \boldsymbol{S}_{c_t}^{(\ell)} \right\rangle_F. \tag{27}$$

Equation (27) reveals the key advantage of the spectral representation: interactions between client gradients and model updates depend on the projected gradient energy inside the update support. Therefore, if the projected gradients of retained and previously forgotten clients have negligible energy on the support allocated to request c_t, the corresponding first-order interactions are negligible.

Proof of Theorem 4.2.

Proof. Let

$$\tilde{\boldsymbol{G}}_q^{(\ell)} = (\boldsymbol{U}_L^{(\ell)})^* \boldsymbol{G}_q^{(\ell)} \boldsymbol{U}_R^{(\ell)}$$

denote the gradient of client q in the spectral basis, evaluated at $\boldsymbol{\theta}^{(t-1)}$. For the one-step projected gradient-ascent update,

$$\boldsymbol{S}_{c_t}^{(\ell)} = \eta P_{\Omega_{c_t}^{(\ell)}} \tilde{\boldsymbol{G}}_{c_t}^{(\ell)}.$$

By local smoothness and $\|\Delta\boldsymbol{\theta}_t\|_F = O(\eta)$, Taylor expansion gives

$$L_q(\boldsymbol{\theta}^{(t)}) - L_q(\boldsymbol{\theta}^{(t-1)}) = \langle g_q, \Delta\boldsymbol{\theta}_t \rangle_F + O(\eta^2).$$

By Eq. equation 27 and the support restriction on $\boldsymbol{S}_{c_t}^{(\ell)}$,

$$\langle g_q, \Delta\boldsymbol{\theta}_t \rangle_F = \eta \sum_{\ell=1}^{L} \left\langle P_{\Omega_{c_t}^{(\ell)}} \tilde{\boldsymbol{G}}_q^{(\ell)}, P_{\Omega_{c_t}^{(\ell)}} \tilde{\boldsymbol{G}}_{c_t}^{(\ell)} \right\rangle_F.$$

For any retained or previously forgotten client q, applying Cauchy–Schwarz within and across layers gives

$$\begin{aligned} \left|\langle g_q, \Delta\boldsymbol{\theta}_t \rangle_F\right| &\le \eta \sum_{\ell=1}^{L} \left\|P_{\Omega_{c_t}^{(\ell)}} \tilde{\boldsymbol{G}}_q^{(\ell)}\right\|_F \left\|P_{\Omega_{c_t}^{(\ell)}} \tilde{\boldsymbol{G}}_{c_t}^{(\ell)}\right\|_F \\ &\le \eta \left\|P_{\Omega_{c_t}} \widetilde{g}_q\right\|_F \left\|P_{\Omega_{c_t}} \widetilde{g}_{c_t}\right\|_F \\ &= \eta \rho_{q,t} \gamma_t \|g_q\|_F \|g_{c_t}\|_F. \end{aligned}$$

Here, the squared norm across layers is the sum of the layerwise squared Frobenius norms. The last equality follows from the definitions of $\rho_{q,t}$ and γ_t and the norm preservation of the unitary transform. Substituting into the Taylor expansion yields

$$\left|L_q(\boldsymbol{\theta}^{(t)}) - L_q(\boldsymbol{\theta}^{(t-1)})\right| \le \eta \rho_{q,t} \gamma_t \|g_q\|_F \|g_{c_t}\|_F + O(\eta^2). \tag{28}$$

For the target client $q = c_t$, the first-order term is

$$\begin{aligned} \langle g_{c_t}, \Delta\boldsymbol{\theta}_t \rangle_F &= \eta \sum_{\ell=1}^{L} \left\|P_{\Omega_{c_t}^{(\ell)}} \tilde{\boldsymbol{G}}_{c_t}^{(\ell)}\right\|_F^2 \\ &= \eta \gamma_t^2 \|g_{c_t}\|_F^2. \end{aligned}$$

Therefore,

$$L_{c_t}(\boldsymbol{\theta}^{(t)}) - L_{c_t}(\boldsymbol{\theta}^{(t-1)}) = \eta \gamma_t^2 \|g_{c_t}\|_F^2 + O(\eta^2).$$

If $\gamma_t > 0$, the leading term is positive, so the target loss increases for sufficiently small $\eta > 0$. □

B.5 FFT as a Practical Instantiation

The above theory applies to any orthonormal basis pair

$$\{\boldsymbol{U}_L^{(\ell)}, \boldsymbol{U}_R^{(\ell)}\}_{\ell=1}^{L}.$$

In practice, we instantiate the framework with Fourier bases. This choice offers three advantages. First, the basis matrices need not be stored explicitly. Second, spectral transforms can be implemented efficiently in

$$O(n \log n)$$

time using `torch.fft`. Third, Fourier representations empirically provide compact energy concentration in transformer layers, making sparse coefficient learning particularly effective.

Therefore, FFT serves as an efficient and memory-free realization of the proposed spectral construction, while all theoretical guarantees remain basis-agnostic.

B.6 Complexity and Communication Analysis

For each layer ℓ, computing the projected gradient

$$(\boldsymbol{U}_L^{(\ell)})^* \boldsymbol{G}^{(\ell)} \boldsymbol{U}_R^{(\ell)}$$

via a 2D FFT requires

$$O(m_\ell n_\ell(\log m_\ell + \log n_\ell))$$

operations, plus

$$O(|\Omega_{c_t}^{(\ell)}|)$$

sparse multiplications for coefficient updates. Summing over all layers gives an overall complexity of order

$$O(d \log d),$$

where d is the total number of model parameters.

The memory cost is determined by the total number of active spectral coefficients:

$$O\left(\sum_{t,\ell} |\Omega_{c_t}^{(\ell)}|\right).$$

In contrast to direct constrained optimization in the full parameter space, which may involve cubic complexity in d, the proposed spectral formulation avoids both dense optimization and explicit storage of the orthonormal bases.

The communication advantage is particularly important in the federated setting. During continual unlearning, only the sparse spectral coefficients

$$\{\boldsymbol{S}_{c_t}^{(\ell)}\}_{\ell=1}^{L}$$

need to be transmitted between clients and the server, rather than dense full-model parameters. This substantially reduces communication overhead and makes the proposed framework well-suited for bandwidth-constrained federated systems.

B.7 Capacity of Orthogonal Spectral Allocation

The exact update-orthogonality guarantee in Lemma 4.1 relies on the existence of disjoint spectral supports across requests. This naturally raises the question of how many continual unlearning requests can be accommodated before spectral supports begin to overlap.

For each layer ℓ, the total number of available spectral coefficients is $m_\ell n_\ell$. If each request occupies $|\Omega_{c_t}^{(\ell)}|$ coefficients on layer ℓ, then the maximum number of fully disjoint requests that can be assigned on that layer is upper bounded by

$$\left\lfloor \frac{m_\ell n_\ell}{|\Omega_{c_t}^{(\ell)}|} \right\rfloor.$$

Aggregating across layers, the total spectral budget is

$$\mathcal{K} = \sum_{\ell=1}^{L} m_\ell n_\ell,$$

which determines the capacity of the framework to maintain exact orthogonality across requests.

Exact update orthogonality holds when the allocated supports are pairwise disjoint at every layer. For a fixed per-request budget $b_\ell > 0$ at each layer, the number of requests that can be assigned disjoint supports is bounded by

$$T_{\max} \leq \min_{\ell} \left\lfloor \frac{m_\ell n_\ell}{b_\ell} \right\rfloor.$$

In particular, C disjoint allocated supports accommodate up to C sequential requests using distinct supports. Support reuse or overlap removes the structural orthogonality guarantee. The first-order functional effect remains governed by the interaction between each update and the relevant client gradients.

Table 4: MIA scores under highly class-skewed partitioning ($\alpha = 0.01$).

Method	MIA Score	Δ_{Retrain}
Retrain	0.28	0.00
FedAvg	0.23	0.05
Exact-Fun	0.08	0.20
FUSED	0.48	0.20
`FedSOUL`	0.17	0.11

Table 5: Balanced MIA accuracy under less class-skewed partitioning ($\alpha = 0.5$). Δ_{Retrain} is reported in percentage points (pp).

Method	MIA Accuracy	Δ_{Retrain} (pp)
Retrain	50.09%	0.00
SFU	56.40%	6.31
FedOSD	53.79%	3.70
FedOrtho	51.15%	1.06
`FedSOUL`	51.42%	1.33

C Membership Inference Evaluation

We evaluate membership inference attacks (MIA) under two levels of client heterogeneity and report the Retrain baseline as a reference. For each method, we calculate its absolute deviation from Retrain:

$$\Delta_{\text{Retrain}} = \left|\text{MIA}_{\text{method}} - \text{MIA}_{\text{Retrain}}\right|. \tag{29}$$

This comparison accounts for the attack behavior of a model retrained without the forgotten data.

As shown in Table 4, the original experiment uses a Dirichlet partition with $\alpha = 0.01$, producing strong client-level class imbalance. In this setting, client removal can resemble class-level unlearning, and low MIA scores require interpretation relative to Retrain.

We further evaluate a Dirichlet partition with $\alpha = 0.5$(Table 5), which reduces class skew and provides a complementary client-level unlearning setting. Here, we report balanced MIA accuracy and assess its proximity to the Retrain reference. The result indicates limited membership distinguishability under the evaluated attack.

D Limitations

Due to limited computational resources, our experiments are currently conducted on moderate-scale benchmark datasets. Evaluating the proposed method on larger-scale and more complex datasets remains an important direction for future work. In addition, `FedSOUL` should be understood as an approximate unlearning method: disjoint spectral supports guarantee update orthogonality, but they do not constitute a certified deletion guarantee and do not by themselves ensure exact functional independence from forgotten data. The functional preservation arguments rely on bounded projected gradient leakage and first-order approximations, which may weaken under highly nonlinear dynamics, repeated requests beyond the available spectral budget, or support allocations with substantial overlap. Finally, the Stage-I retained-adapter initialization introduces an upfront cost that should be considered when deploying the method in settings with very few unlearning requests.

E Broader impacts

This work supports the right to be forgotten by enabling efficient sequential data removal in federated systems, with direct benefits for privacy compliance (GDPR, CCPA). A potential negative impact is that selective unlearning could be misused to erase accountability-relevant knowledge or degrade model utility on targeted subpopulations. We discuss these limitations and the approximate nature of our guarantees in the Limitations section.

NeurIPS Paper Checklist

i. **Claims**

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [Yes]

Justification: The abstract and introduction clearly state the three main contributions: problem formalization, the FedSOUL framework with spectral orthogonality, and empirical evaluation. Theoretical results are scoped to first-order approximations and bounded leakage, and limitations of the approximate unlearning guarantee are acknowledged in the Limitations section.

Guidelines:

- The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

ii. **Limitations**

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: Section D discusses the limitation.

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

iii. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Please refer to Section 4.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

iv. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The experimental details have been explained in the experiment section.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

v. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We submit our code as supplement material.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

vi. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: Coding provides full implementation details.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

vii. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: All these have been explained in the experiment section and the appendix

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

viii. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: These have been explained in the experiment section and the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

ix. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: The research conducted in the paper conform and NeurIPS code of ethics.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

x. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Section E shows the impacts

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

xi. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [Yes]

Justification: The model has been well safeguarded.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

xii. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The original datasets and references have been properly cited in this paper.1016

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

xiii. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The new asset of this paper is the proposed model. It has been well documented and detailed in the paper.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

xiv. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

xv. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

xvi. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: We just use llm to edit the paper.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.