Federated Unlearning with Gradient Adaptive Shaping

Siyin Huang* **Yufan Liao*** **Yuelin Du*** **Yu Zhang†**
Southern University of Science and Technology
{huangsy2024, 12210626, 12211925}@mail.sustech.edu.cn yu.zhang.ust@gmail.com

Abstract

Federated unlearning aims to efficiently remove the influence of a specific client from a trained global model in federated learning without full retraining. However, existing approaches struggle to preserve the utility of the remaining clients because unlearning can be unstable at both the client and server stages. Local forgetting updates may become overly aggressive and destabilize client side optimization, while server aggregation can amplify conflicts between forgetting updates and retained client updates. To address both failure modes within a unified framework, we propose Federated Unlearning with Gradient Adaptive Shaping (FUGAS), a history-free gradient shaping framework that stabilizes the entire unlearning pipeline. On the unlearning client side, we employ a bounded preference objective that utilizes the pre-unlearning model predictions on the forgetting data as negative references to controllably steer the model away from unlearning client knowledge while requiring only minimal storage for reference caches rather than full historical updates. On the server side, we introduce a compatibility projection mechanism that reshapes the aggregated unlearning update to remain compatible with directions estimated from retained clients. Our analysis shows that FUGAS promotes stability and ensures non-increasing empirical risk on retained distributions while establishing an excess risk bound relative to retraining. Extensive experiments demonstrate that FUGAS achieves effective unlearning while consistently maintaining high accuracy on retained data.

1 Introduction

Federated unlearning [1–3] has become a fundamental requirement in practical federated learning systems [4–8], where a trained global model must remove the influence of designated clients while preserving the utility of the retained clients. Ideally, the unlearned model should behave as if the forgetting client had never participated in training and should approximate the model obtained by retraining only on the retained clients. However, full retraining is prohibitive in federated learning because it requires restarting distributed optimization over many clients and communication rounds.

A common strategy for reducing retraining cost is to store historical client updates or intermediate checkpoints during training and reuse them after an unlearning request [9–11]. Such history-based methods can reconstruct or calibrate the global model by replaying stored training information, but their memory cost grows with the number of communication rounds and participating clients. To alleviate this storage bottleneck, recent work turns to history-free paradigms that directly update model parameters through optimization on the data requested for removal, without relying on stored server-side histories. In this work, memory efficiency refers to eliminating server-side historical updates and checkpoints rather than eliminating all auxiliary local quantities.

*Equal contribution.
†Corresponding author.

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

The removal of server-side history makes federated unlearning substantially more memory efficient, but it also introduces two distinct instability mechanisms that jointly undermine retained utility. On the unlearning client-side, techniques [12–14] that maximize the loss on the data to be forgotten frequently suffer from gradient explosion because unbounded objectives drive model parameters into unstable regions, leading to drastic parameter changes. Beyond local instability, on the server side, the unlearning process is further compromised by destructive gradient interference during global aggregation [13, 15–17]. The gradient updates derived from the forgetting data often lie in direct opposition to the optimization directions required by the retained clients, resulting in utility degradation. These two failure modes are coupled and jointly degrade the utility of the global model.

To address both coupled failure modes, we propose Federated Unlearning with Gradient Adaptive Shaping (FUGAS), a history-free gradient shaping framework designed to improve stability and preserve utility throughout the unlearning pipeline without storing historical updates. During the client-side unlearning phase, FUGAS performs a preference optimization task by treating the pre-unlearning model predictions on the data to be forgotten as negative references, which steers the model away from unlearning client knowledge while keeping update magnitudes controlled. During the server-side aggregation phase, FUGAS applies a compatibility projection mechanism that reshapes the unlearning update onto a subspace estimated from retained client gradients, which reduces destructive interference and preserves the utility of the retained clients. We further provide theoretical analysis to clarify why the proposed shaping mechanism improves stability and retained utility. We also establish an excess risk bound relating the final unlearned model to the gold standard retraining solution on retained clients. Empirically, we consistently observe strong forgetting efficacy together with high retained accuracy across benchmarks and unlearning configurations.

In summary, our work makes three key contributions. First, we identify two coupled instability mechanisms in history-free federated unlearning and propose FUGAS, a framework that jointly addresses client-side bounded forgetting and retained gradient compatibility within a unified gradient shaping pipeline. Second, we provide theoretical analysis characterizing gradient stability and non-increasing retained empirical risk under standard smoothness conditions, together with an excess-risk bound that relates the unlearned model to the gold-standard retrained solution. Third, we conduct extensive experiments across diverse settings, showing that FUGAS effectively removes the influence of the clients requested for removal while maintaining high accuracy on retained clients.

2 Related Work

Machine Unlearning. Machine unlearning seeks to efficiently remove the influence of designated data from trained models so that the resulting model behaves as if it were trained without it [18–21]. Full retraining from scratch remains the gold standard, but its cost is prohibitive. To mitigate this, exact unlearning methods [22, 23] like SISA [18] partition the dataset into isolated shards, enabling targeted retraining of only the affected submodels at the expense of substantial storage and recomputation. Approximate unlearning methods [24–28] seek greater efficiency by directly modifying the trained model's parameters to approximate a retrained state. Representative techniques include reversing the learning process through gradient ascent [29–31], leveraging influence functions to estimate data impact [32, 33], and employing knowledge distillation to suppress targeted information [34–38]. However, these centralized mechanisms rely on unrestricted access to the complete dataset to calibrate the trade-off between forgetting and retaining. This assumption is violated in federated learning, where data remain decentralized under privacy constraints and client data distributions can be highly heterogeneous. Consequently, directly applying centralized unlearning techniques to the federated setting is infeasible, as they lack the global visibility required to resolve the inevitable conflicts between unlearning local data and maintaining global model utility.

Federated Unlearning. Federated unlearning [39–41, 14, 3] aims to erase a forgetting client's impact from the global model while approximating the effect of full retraining without prohibitive cost. Early work achieves this by following an update-storage paradigm [42–44], which replays or calibrates stored client updates as in FedEraser [1] and FedRecovery [45]. Although subsequent variants [46, 47, 44] attempt to reduce overhead through selective retention, compression, or knowledge distillation, they rely on substantial server-side history, making them impractical in federated settings where storage scales linearly with training rounds. To eliminate this storage dependency, recent research explores history-free solutions [48, 9, 13, 49] that formulate unlearning as an optimization

problem defined on the data requested for removal to reverse prior updates. This line of work shifts the focus from historical replay to direct update reversal, yet this shift also introduces new stability challenges across the unlearning pipeline. Recent methods [12, 50, 51] improve stability through client-side unlearning objectives and server-side orthogonality projection. Nevertheless, strict orthogonality may discard gradient components that are non-opposing and potentially useful for utility preservation. In this work, we advance this line of research by providing a unified treatment of failure modes at both the client and server stages. FUGAS introduces a bounded preference objective on the client side, coupled with a flexible compatibility projection on the server side. This synergistic design eliminates the storage overhead of prior paradigms, stabilizes the federated unlearning process and balances effective forgetting with utility preservation.

3 Preliminaries

Federated Learning. We consider a federated learning system with K clients indexed by $k \in [K] \triangleq \{1, \ldots, K\}$, where each client k possesses a private dataset $\mathcal{D}_k$. The goal is to optimize a global model θ^* that minimizes a weighted average of local loss, $\theta^* = \arg\min_\theta \sum_{k=1}^K p_k \mathcal{L}(\mathcal{D}_k; \theta)$. Here, $p_k \geq 0$ represents the importance weight for client k, typically proportional to the dataset size, satisfying the constraint $\sum_{k=1}^K p_k = 1$. In each round t, the k-th client starts from the current global model θ^t and performs local optimization to obtain an updated model θ_k^{t+1}. The server subsequently aggregates these local updates to compute the new global model $\theta^{t+1} = \theta^t - \eta_g \sum_{k=1}^K p_k(\theta^t - \theta_k^{t+1})$, where η_g is the server-side global learning rate. Following the standard setting in federated learning, we assume a trusted server that honestly executes the aggregation protocol, while client data remain decentralized and are not shared with the server.

Federated Unlearning. Federated unlearning seeks to excise the impact of the forgetting clients $F \subseteq [K]$ from the trained model while preserving the knowledge of the retained clients $R = [K] \setminus F$. The lifecycle of federated unlearning consists of a pre-unlearning phase and an unlearning phase. We refer to the standard FL training that produces θ^0 as the pre-unlearning phase. After an unlearning request, the system enters the unlearning stage. In each unlearning round, retained clients continue the standard local training on $\mathcal{D}_k$ for $k \in R$, while forgetting clients $k \in F$ perform local updates designed to remove their contribution. A key challenge in this phase is that certain approaches require the server to cache per-round model checkpoints or client updates to facilitate unlearning, inducing a storage overhead that grows linearly with both the number of training rounds and the number of participating clients. In this work, we adopt a history-free setting where the server retains no such historical trajectories, operating solely on information available in the current unlearning round. The trusted server then processes and aggregates the received updates to compute the next global model.

Gradient Conflict. Let g_F^t and g_R^t denote the aggregated gradients derived from the forgetting clients and the retained clients at round t, respectively. Ideally, unlearning should modify the model to increase the loss on forgetting data without affecting the performance on retained data. However, gradient conflict arises when the optimization directions of F and R are contradictory, quantified by a negative inner product $\langle g_F^t, g_R^t \rangle < 0$. This condition implies that the gradient intended to erase the knowledge from F forms an obtuse angle with the gradient required to maintain performance on R, pushing the objectives of forgetting and retention in conflicting directions.

4 Methodology

To achieve effective unlearning without compromising the utility of the global model, FUGAS operates as a history-free gradient shaping framework. As depicted in Figure 1, we first stabilize the client-side optimization by using a bounded optimization objective, and subsequently mitigate server-side aggregation interference by enforcing compatibility with retained-client update directions. We will introduce the two parts in the following sections.

4.1 Client-Side Training

During unlearning round t, each client k starts from the current global model θ^t and performs local training. For a forgetting client $k \in F$, we formulate unlearning as a preference optimization task.

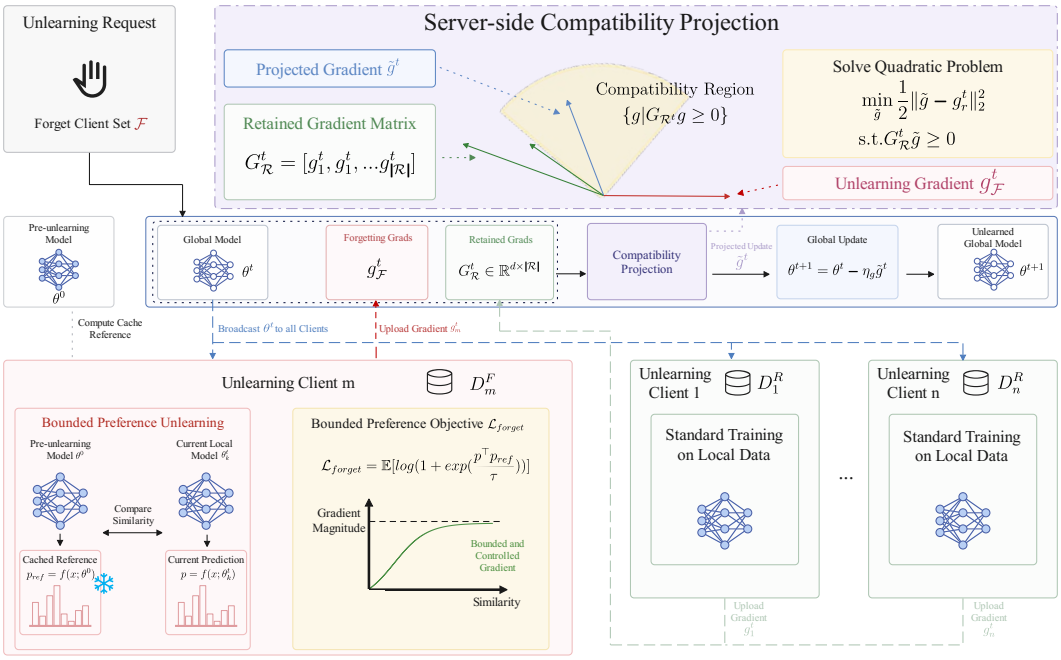


Figure 1: Overview Pipeline of FUGAS.

We treat the prediction of the pre-unlearning global model θ^0 as a negative reference that a unlearning model should diverge from.

At the beginning of the unlearning phase, client k computes and caches the fixed reference probability distribution $\mathbf{p}^{\text{ref}} = f(x;\theta^0)$ for each sample $x \in \mathcal{D}_k$. On unlearning round t, the client then optimizes the current model θ^t_k to produce an output distribution $\mathbf{p} = f(x;\theta^t_k)$ that exhibits a low similarity to $\mathbf{p}^{\text{ref}}$. Hence, we formulate the objective function of a forgetting client as

$$\mathcal{L}_{\text{forget}}(\mathcal{D}_k;\theta^t_k) = \mathbb{E}_{x\in\mathcal{D}_k}\left[\log\left(1 + \exp\left(\frac{\mathbf{p}^\top \mathbf{p}^{\text{ref}}}{\tau}\right)\right)\right], \tag{1}$$

Meanwhile, a retained client $k \in R$ performs standard local training to preserve utility on its data, and computes a retained gradient g^t_k that will be used to guide aggregation-time shaping on the server. This allows the server to estimate the retained update directions in each unlearning round.

4.2 Server-Side Aggregation

While the local objective ensures stability, the challenge of gradient conflict persists during aggregation. Owing to the conflict between the unlearning gradients and the gradients of retained clients, simply aggregating these gradients incurs destructive interference. To resolve this, we employ a gradient projection mechanism at the server.

The server first forms the aggregated forgetting gradient $g^t_F = \sum_{k\in F} p_k\, g^t_k$, which summarizes the removal signal from potentially multiple forgetting clients. Simultaneously, the server collects retained gradients and forms a matrix $\mathbf{G}^t_R \in \mathbb{R}^{|R|\times d}$, where $|R|$ is the number of retained clients and each row corresponds to a specific retained gradient g^t_k. Then the server computes a compatible update direction $\tilde{g}^t$ that stays close to g^t_F while remaining non-opposing to retained directions. We formulate this as the following constrained primal optimization problem:

$$\tilde{g}^t = \arg\min_{\tilde{g}\in\mathbb{R}^d} \frac{1}{2}\|\tilde{g} - g^t_F\|^2_2 \quad \text{s.t.} \quad \mathbf{G}^t_R\tilde{g} \geq \mathbf{0}. \tag{2}$$

The constraint $\mathbf{G}^t_R\tilde{g} \geq \mathbf{0}$ ensures that the final update does not oppose the optimization direction of any retained client, effectively protecting their specific knowledge from erasure. This helps preserve the model utility.

Solving Eq. (2) directly is computationally expensive due to the high dimensionality d of the model parameters. To ensure scalability, we solve the equivalent dual problem. By introducing Lagrange multipliers $\mathbf{v} \in \mathbb{R}^{|R|}$, we can transform the high-dimensional primal problem into a low-dimensional dual quadratic program over $\mathbf{v}$ as

$$\min_{\mathbf{v} \geq \mathbf{0}} \ \frac{1}{2}\mathbf{v}^\top \mathbf{P}\mathbf{v} + \mathbf{q}^\top \mathbf{v}, \tag{3}$$

where $\mathbf{P} = \mathbf{G}_R^t {\mathbf{G}_R^t}^\top$ and $\mathbf{q} = \mathbf{G}_R^t g_F^t$. The dual optimizes over $|R|$ retained clients rather than d model parameters and is substantially cheaper when $|R| \ll d$. Given the optimal solution $\mathbf{v}^*$, the compatible forgetting update is recovered as

$$\tilde{g}^t = g_F^t + {\mathbf{G}_R^t}^\top \mathbf{v}^*. \tag{4}$$

We provide the detailed derivation in Appendix A.

Once the optimal effective gradient $\tilde{g}^t$ is computed, the server updates the global model as $\theta^{t+1} \leftarrow \theta^t - \eta_g \tilde{g}^t$. This update preserves the forgetting update as much as possible while preventing harmful opposition to retained client optimization, thereby supporting stable global unlearning.

4.3 Analysis

Bounded Forgetting Gradients. Methods based on gradient ascent forget by maximizing an unbounded loss on the data requested for removal, so the update magnitude can grow without limit. In contrast, our preference-based objective yields bounded forgetting gradients under a bounded Jacobian assumption. We formalize this property below.

Assumption 1. *Let $\mathbf{p} = f(x;\theta)$ denote the model output distribution and assume that its parameter Jacobian is bounded by $\|\nabla_\theta \mathbf{p}\|_2 \leq L$.*

Proposition 1. *Under Assumption 1, the parameter gradient of the forgetting objective $\mathcal{L}_{\text{forget}}$ defined in Eq.* (1) *has a bounded ℓ_2 norm for any temperature $\tau > 0$.*

Discussion. Proposition 1 shows that the forgetting gradient stays bounded throughout unlearning. The bound is distribution-agnostic, so it holds for any client data distribution including non-IID partitions. The temperature τ acts as an explicit regulator of the update magnitude. A larger τ induces a smoother loss landscape with smaller gradient steps, while a smaller τ allows for more aggressive updates. The more detailed proof is in Appendix B.1.

Utility Preservation via Gradient Projection. Proposition 1 establishes bounded forgetting gradients, while the forgetting direction g_F^t may still conflict with retained-client gradients during server aggregation. FUGAS mitigates such conflicts by projecting g_F^t onto a compatibility region defined by retained-client gradients. In the following, we provide an analysis to show how this projection prevents utility degradation.

Proposition 2. *Let $\tilde{g}^t$ be the update computed by the FUGAS projection mechanism in Eq.* (2)*. For the global update $\theta^{t+1} = \theta^t - \eta_g \tilde{g}^t$ with $\eta_g > 0$, the compatibility constraints guarantee non-increasing empirical risk on the retained clients under a first-order Taylor approximation.*

Discussion. Proposition 2 indicates compatibility projection preserves retained utility during unlearning without requiring convexity or convergence of the pre-unlearning model. If the model is not converged, FUGAS ensures that the unlearning update contains a component that acts as further training for the retained clients. Thus, FUGAS utilizes the projection to excise the influence of the forgetting clients while maintaining or potentially even improving the utility for remaining clients. The more detailed proof is in Appendix B.2.

Retained Excess Risk Relative to Retraining. Proposition 2 establishes first-order compatibility with retained-client gradients, we further analyze the retained excess risk of the unlearned model θ^T relative to the retrained model θ_R^*.

Assumption 2. *The retained objective $\mathcal{L}_R(\theta)$ satisfies the μ-PL condition and is L-smooth. The forgetting objective has gradients bounded by G_F, and the projected updates satisfy $\|\tilde{g}^t\| \leq G$.*

Table 1: Unlearning efficacy and retained accuracy CIFAR-10 and CIFAR-100 benchmarks. All values are percentages, with accuracy reported as mean ± standard deviation. $|F|$ denotes the number of forgetting clients. The highest retained accuracy among unlearning methods is bolded.

Method	CIFAR-10 non-IID		CIFAR-10 IID		CIFAR-100 non-IID		CIFAR-100 IID	
	ASR↓	Acc↑	ASR↓	Acc↑	ASR↓	Acc↑	ASR↓	Acc↑
$\|F\| = 1$								
Pre-Unlearning θ^0	99.8	$62.5_{\pm 1.0}$	99.7	$81.2_{\pm 1.6}$	97.5	$41.4_{\pm 5.0}$	99.2	$49.3_{\pm 1.0}$
Retraining	1.3	$71.2_{\pm 0.7}$	1.7	$80.9_{\pm 0.3}$	0.0	$50.1_{\pm 0.1}$	0.0	$52.9_{\pm 0.5}$
GA	0.0	$9.7_{\pm 0.0}$	0.0	$9.9_{\pm 0.2}$	0.0	$1.0_{\pm 0.0}$	0.0	$0.8_{\pm 0.0}$
PGD	16.3	$28.9_{\pm 1.1}$	4.6	$30.6_{\pm 2.0}$	12.3	$25.1_{\pm 0.1}$	0.0	$38.9_{\pm 0.5}$
FedBU	0.0	$21.4_{\pm 0.1}$	0.0	$59.4_{\pm 0.9}$	0.0	$20.1_{\pm 1.3}$	11.2	$42.6_{\pm 0.2}$
FedEraser	5.6	$27.6_{\pm 0.2}$	0.1	$48.4_{\pm 0.7}$	0.2	$25.0_{\pm 0.2}$	0.4	$37.2_{\pm 1.0}$
FedOSD	0.0	$46.7_{\pm 1.3}$	0.0	$71.7_{\pm 0.1}$	0.0	$47.5_{\pm 1.1}$	0.0	$47.2_{\pm 0.9}$
RFUL	0.4	$48.8_{\pm 0.3}$	0.0	$67.8_{\pm 0.0}$	0.0	$47.5_{\pm 0.1}$	5.3	$41.7_{\pm 0.4}$
NoT	0.0	$42.3_{\pm 1.2}$	8.7	$75.3_{\pm 0.3}$	3.8	$39.9_{\pm 0.5}$	4.2	$48.5_{\pm 0.3}$
FUGAS (Ours)	0.1	$\mathbf{67.2}_{\pm 0.0}$	0.0	$\mathbf{78.7}_{\pm 0.0}$	0.0	$\mathbf{50.9}_{\pm 0.2}$	0.0	$\mathbf{51.3}_{\pm 0.1}$
$\|F\| = 2$								
Pre-Unlearning θ^0	99.9	$41.9_{\pm 0.9}$	99.9	$72.8_{\pm 0.7}$	99.8	$28.6_{\pm 1.2}$	99.9	$38.9_{\pm 1.3}$
Retraining	0.8	$79.8_{\pm 0.1}$	0.2	$77.6_{\pm 0.1}$	0.0	$51.3_{\pm 0.6}$	0.0	$48.7_{\pm 0.2}$
GA	0.0	$2.3_{\pm 0.1}$	0.0	$1.0_{\pm 0.1}$	0.0	$1.2_{\pm 0.2}$	0.0	$1.3_{\pm 0.3}$
PGD	0.0	$35.9_{\pm 0.1}$	0.5	$64.1_{\pm 0.3}$	1.3	$26.7_{\pm 0.6}$	2.1	$36.6_{\pm 0.2}$
FedBU	0.0	$42.8_{\pm 0.2}$	0.2	$59.3_{\pm 0.1}$	0.0	$36.1_{\pm 1.0}$	0.4	$32.1_{\pm 0.7}$
FedEraser	0.0	$28.6_{\pm 1.2}$	5.1	$57.2_{\pm 0.5}$	0.1	$20.3_{\pm 0.6}$	0.1	$23.4_{\pm 0.1}$
FedOSD	0.0	$37.2_{\pm 0.1}$	0.0	$64.7_{\pm 2.1}$	0.3	$38.6_{\pm 0.0}$	0.0	$40.9_{\pm 1.5}$
RFUL	3.2	$42.0_{\pm 0.0}$	0.0	$60.1_{\pm 0.2}$	4.2	$35.1_{\pm 0.0}$	5.0	$39.2_{\pm 0.4}$
NoT	0.0	$44.7_{\pm 0.9}$	6.3	$66.2_{\pm 0.1}$	0.0	$41.7_{\pm 1.4}$	0.0	$36.9_{\pm 0.6}$
FUGAS (Ours)	0.0	$\mathbf{59.8}_{\pm 0.2}$	0.0	$\mathbf{70.3}_{\pm 0.7}$	0.0	$\mathbf{44.6}_{\pm 1.2}$	0.0	$\mathbf{48.5}_{\pm 0.0}$

Proposition 3. *Let $\alpha = \sum_{k \in F} p_k \in (0, 1)$ denote the original training weight of the forgetting clients. Under Assumption 2, after T unlearning rounds with learning rate $\eta_g > 0$, the retained excess risk satisfies*

$$\mathcal{L}_R(\theta^T) - \mathcal{L}_R(\theta_R^*) \leq \frac{\alpha^2 G_F^2}{2\mu(1-\alpha)^2} + \frac{TL\eta_g^2 G^2}{2}. \tag{5}$$

Discussion. The quantity α is not used by FUGAS and only characterizes how much the forgetting clients contributed during the original training. It appears in the bound as an unavoidable initial drift term, while the unlearning procedure contributes only the $T\eta^2$ accumulation term under bounded and compatible updates. The more detailed proof is in Appendix B.3.

5 Experiment

5.1 Experiment Setup

Baselines. We compare FUGAS with other federated unlearning methods. Retraining denotes training the global model from scratch on the retained clients. Gradient ascent (GA) directly maximizes the loss on the forgetting data without constraints. FedEraser [1] reconstructs the model by using historical updates. PGD [9] maximizes the loss on the forgetting client while constraining the model within an ℓ_2-norm ball of the pre-unlearning parameters. FedBU [13] bounds forgetting information while preserving parameters important to retained accuracy. FedOSD [50] computes an orthogonal steepest-descent direction to mitigate gradient conflict. NoT [49] unlearns by negating weights in first layer. RFUL [51] retains model performance via knowledge distillation.

Table 2: Ablation study of preference selection strategies on CIFAR-10 and CIFAR-100 benchmarks. All values are percentages, with accuracy reported as mean ± standard deviation. The highest retained accuracy is bolded.

Preference Selection	CIFAR-10 non-IID		CIFAR-10 IID		CIFAR-100 non-IID		CIFAR-100 IID	
	ASR ↓	Acc ↑	ASR ↓	Acc ↑	ASR ↓	Acc ↑	ASR ↓	Acc ↑
FUGAS + Uniform	0.0	$66.1_{\pm 0.3}$	1.3	$78.0_{\pm 0.0}$	0.0	$49.1_{\pm 0.1}$	0.0	$50.9_{\pm 0.2}$
FUGAS + Augmentation	0.1	$66.8_{\pm 0.9}$	0.1	$72.1_{\pm 0.6}$	0.0	$48.3_{\pm 0.0}$	0.2	$50.5_{\pm 0.8}$
FUGAS (Hard Label)	0.2	$65.9_{\pm 1.2}$	4.0	$77.3_{\pm 0.2}$	0.0	$49.4_{\pm 0.1}$	0.0	$48.7_{\pm 0.0}$
FUGAS (Ours)	0.1	$\mathbf{67.2}_{\pm 0.0}$	0.0	$\mathbf{78.7}_{\pm 0.0}$	0.0	$\mathbf{50.9}_{\pm 0.2}$	0.0	$\mathbf{51.3}_{\pm 0.1}$

Datasets and Models. We evaluate on CIFAR-10 and CIFAR-100 using a ResNet-18 under both IID and non-IID data distributions. In the IID setting, we randomly shuffle the full dataset and uniformly split it across clients. In the non-IID setting, we sample class label proportions from a Dirichlet distribution with a concentration parameter of 0.1. To reflect practical local evaluation, we construct the local test dataset for each client to match the distribution of its corresponding local training data.

Implementation Details. The process consists of two phases, including 500 rounds of pre-training, followed by 10 rounds of unlearning. The federated setting consists of ten clients with full participation in every communication round, and one or two clients are chosen for unlearning. We perform a grid search for each baseline to identify its optimal learning rate within the range $[10^{-5}, 10^{-2}]$, reporting the best performance achieved. Furthermore, we set the global learning rate to 1 for all methods in the server-side aggregation. For FUGAS, we set the temperature parameter to $\tau = 3$. We assess utility using test accuracy on the test dataset of retained clients. To quantify unlearning effectiveness, we employ two attack-based metrics [52, 53] including Attack Success Rate (ASR) and Membership Inference Attack (MIA) accuracy. The backdoor attack is configured with a fixed 7×7 pixel trigger injected into 100% of the forgetting clients' training data. During pre-training, the global model is fully compromised by this trigger, achieving an initial ASR of nearly 100%. All results are averaged over three random seeds. Further details are provided in Appendix D.1.

5.2 Results

We report unlearning efficacy measured by ASR and retained classification accuracy in Table 1. We evaluate FUGAS under more complex federated unlearning scenarios in Appendix D.2. We also report sensitivity analysis of temperature in Appendix D.3. More analyses regarding training overhead can be found in the Appendix D.8.

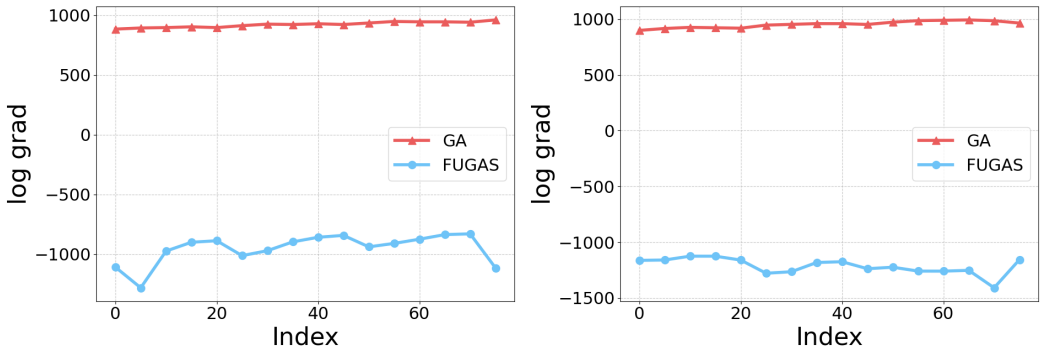
Table 1 shows that Retraining achieves strong retained accuracy with negligible ASR at the cost of full retraining. At the other extreme, Gradient Ascent successfully reduces ASR to zero but suffers from catastrophic forgetting, causing the accuracy on remaining clients to collapse to near-random levels. This confirms the instability of unbounded loss maximization. Similarly, PGD and FedBU effectively erase the forgetting data but incur severe utility degradation. While FedEraser and FedOSD offer a better balance by reducing ASR with moderate utility loss, they still exhibit a significant accuracy gap compared to the Retraining baseline, particularly in non-IID settings and when multiple clients request unlearning. In contrast, FUGAS achieves a more favorable forgetting–utility trade-off. It consistently drives ASR to near-zero levels while maintaining high retained accuracy that is closest to Retraining among memory-efficient methods. These improvements align with the design of FUGAS.

5.3 Ablation Study

Effectiveness of Preference Selection Strategies. We assess different preference signal designs by comparing FUGAS against three variants. The first two baselines transform diverging from the negative reference while maximizing similarity with a positive reference. FUGAS + Uniform sets the positive reference as a uniform probability vector to enforce maximum output uncertainty on the forgetting data, and FUGAS + Augmentation utilizes the predictions of pre-unlearning model on data-augmented views of the forgetting samples as positive references. FUGAS (Hard Label) replaces the soft prediction reference with the hard ground-truth labels of the forgetting data.

Table 3: Ablation study of gradient conflict mitigation strategies on CIFAR-10 and CIFAR-100 benchmarks. All values are percentages, with accuracy reported as mean ± standard deviation. The highest retained accuracy is bolded.

Mitigation Strategy	CIFAR-10 non-IID		CIFAR-10 IID		CIFAR-100 non-IID		CIFAR-100 IID	
	ASR ↓	Acc ↑	ASR ↓	Acc ↑	ASR ↓	Acc ↑	ASR ↓	Acc ↑
FUGAS w/o Projection	0.0	$54.0_{\pm 0.1}$	0.0	$73.8_{\pm 0.5}$	0.0	$28.5_{\pm 0.6}$	0.0	$46.8_{\pm 0.1}$
FUGAS w/ Orthogonal	1.9	$45.8_{\pm 0.3}$	2.8	$74.6_{\pm 1.2}$	0.4	$46.8_{\pm 0.2}$	0.0	$47.9_{\pm 0.5}$
FUGAS (Ours)	0.1	$\mathbf{67.2}_{\pm 0.0}$	0.0	$\mathbf{78.7}_{\pm 0.0}$	0.0	$\mathbf{50.9}_{\pm 0.2}$	0.0	$\mathbf{51.3}_{\pm 0.1}$



(a) CIFAR-10 (b) CIFAR-100

Figure 2: Comparison of gradient magnitudes for our objective versus a cross-entropy maximization objective on CIFAR-10 and CIFAR-100. The y-axis plots the log of the gradient norm.

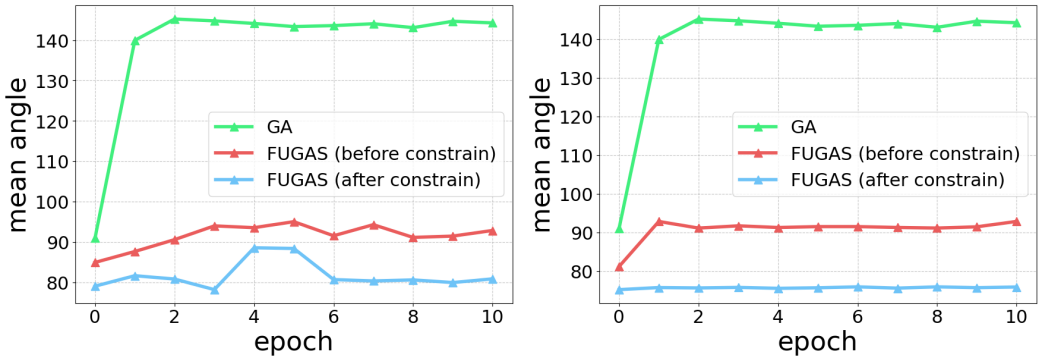
The results, summarized in Table 2, reveal distinct trade-offs. We observe that introducing positive preferences yields comparable retained accuracy but consistently results in higher ASR. This suggests that the auxiliary positive objective creates optimization conflicts, preventing the model from fully excising the forgetting knowledge. Furthermore, calculating positive reference incurs additional computational costs. Conversely, using hard labels as the negative reference proves overly aggressive. It enforces a coarse-grained, class-level forgetting that distorts feature representations shared with retained tasks. Overall, our proposed FUGAS achieves the optimal balance. By focusing solely on instance-specific divergence from soft labels, it attains near-zero ASR and preserves the highest retained accuracy, without inducing the damage associated with hard labels or the gradient interference caused by auxiliary positive objectives.

Effectiveness of Gradient Projection Mechanism. We quantify the effect of our projection strategy during the server-side aggregation. We compare FUGAS against a baseline without projection and a strict orthogonality constraint, as detailed in Table 3.

Applying our preference loss without projection stabilizes the optimization and recovers significant utility, yet performance remains suboptimal due to unresolved gradient conflicts. Introducing a strict orthogonality constraint improves upon the unconstrained baseline but degrades performance compared to FUGAS in several settings. This observation confirms that strict orthogonality is overly restrictive, as it discards update components that are compatible with the retained task but not perfectly perpendicular. By enforcing a flexible compatibility constraint, FUGAS effectively mitigates destructive interference while preserving the maximal amount of shared gradient information, thereby achieving the highest retained accuracy while maintaining near-zero ASR.

5.4 Analysis

Analysis of Gradient Stability. We compare our bounded preference objective with a naive cross-entropy (CE) maximization baseline by inspecting the ℓ_2-norm of the gradients for each layer. The gradients are computed on the forgetting dataset at the end of the unlearning phase, as illustrated in Figure 2. We observe that CE maximization produces gradients that are orders of magnitude



(a) CIFAR-10 non-IID (b) CIFAR-100 non-IID

Figure 3: Visualization of our gradient projection mechanism in mitigating gradient conflict.

Table 4: Comparison of FUGAS using individual and aggregated retained gradients on CIFAR-10 and CIFAR-100 benchmarks. All values are percentages, with accuracy reported as mean ± standard deviation. The highest retained accuracy is bolded.

Method	CIFAR-10 non-IID		CIFAR-10 IID		CIFAR-100 non-IID		CIFAR-100 IID	
	ASR ↓	Acc ↑	ASR ↓	Acc ↑	ASR ↓	Acc ↑	ASR ↓	Acc ↑
Aggregated FUGAS	0.0	**67.3**$_{\pm 0.1}$	0.0	77.6$_{\pm 0.1}$	0.0	**51.0**$_{\pm 0.3}$	0.0	51.1$_{\pm 0.0}$
Original FUGAS	0.3	67.2$_{\pm 0.2}$	0.0	**78.2**$_{\pm 0.2}$	0.0	50.9$_{\pm 0.0}$	0.0	**51.3**$_{\pm 0.0}$

larger across layers than our objective, consistent with the severe retained-utility degradation of GA unlearning in Table 1. This gradient explosion phenomenon directly explains the accuracy decrease since such drastic updates destructively interfere with learned representations. In contrast, our bounded objective ensures that the gradient magnitudes remain small and controlled across all layers. This stability prevents numerical divergence, allowing the model to retain substantial utility even before the server-side gradient projection is applied, as evidenced by the robust performance of the "FUGAS w/o Projection" results in Table 3.

Analysis of Gradient Conflict. We further investigate gradient conflict by measuring the angle between the forgetting update and the retained direction during server-side aggregation phase, as visualized in Figure 3. With CE maximization, the forgetting direction frequently forms an obtuse angle with the retained aggregate gradient, indicating strong conflict and explaining the observed utility loss in Table 1. Interestingly, our preference-based objective alone significantly alleviates this collapse by generating gradients that are less diametrically opposed to the retained task. This natural alignment is then rigorously enforced by our gradient projection mechanism, which projects the unlearning update into the compatibility region. By the proposed projection, FUGAS mathematically resolves the conflict, allowing the model to traverse a parameter trajectory that satisfies the unlearning requirement without ascending the loss surface of the retained clients.

Analysis with Secure Aggregation. Original FUGAS requires individual retained-client gradients to enforce compatibility constraints. To support secure aggregation, we introduce a variant that replaces these constraints with a single constraint on the aggregated retained gradient $\bar{g}_R^t = \sum_{k \in R} p_k g_k^t$. The resulting projection admits a closed-form solution. This variant requires only the aggregated retained gradient and avoids solving a quadratic program. As shown in Table 4, it achieves comparable forgetting effectiveness and retained accuracy to the original FUGAS, demonstrating compatibility with secure aggregation without exposing individual retained-client gradients. The per-client constraints bring a small gain at the cost of exposing individual gradients, and we can choose between the two versions according to our privacy requirement. Since the aggregate itself may still leak information about client data, Appendix D.7 further shows that FUGAS remains effective when retained updates are privatized with the Gaussian mechanism.

Table 5: Analysis of knowledge interference on CIFAR-10 and CIFAR-100 benchmarks. F-Acc, O-Acc, and R-Acc measure accuracy on forgetting-specific, overlapping, and retained-specific classes, respectively. All values are percentages, reported as mean ± standard deviation. The highest O-Acc and R-Acc among unlearning methods are bolded.

Method	CIFAR-10			CIFAR-100		
	F-Acc ↓	O-Acc ↑	R-Acc ↑	F-Acc ↓	O-Acc ↑	R-Acc ↑
Retraining	$0.0_{\pm 0.0}$	$32.8_{\pm 0.5}$	$76.6_{\pm 0.3}$	$0.2_{\pm 0.0}$	$20.7_{\pm 1.3}$	$50.1_{\pm 0.1}$
FedEraser	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$69.3_{\pm 1.3}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$25.7_{\pm 0.3}$
FedOSD	$0.1_{\pm 0.1}$	$0.1_{\pm 0.1}$	$72.6_{\pm 1.2}$	$1.5_{\pm 0.6}$	$4.7_{\pm 0.3}$	$50.3_{\pm 0.2}$
FUGAS (Ours)	$0.1_{\pm 0.2}$	$\mathbf{17.7}_{\pm 0.8}$	$\mathbf{77.4}_{\pm 0.5}$	$2.6_{\pm 0.5}$	$\mathbf{12.9}_{\pm 0.4}$	$\mathbf{51.4}_{\pm 0.9}$

5.5 Knowledge Interference

To assess whether unlearning removes client-specific information without harming shared representations, we report accuracy on three knowledge partitions. We denote F-Acc as accuracy on classes that appear only on forgetting clients, O-Acc as accuracy on classes shared by forgetting and retained clients, and R-Acc as accuracy on classes that appear only on retained clients. We use a controlled split on CIFAR-10 and CIFAR-100. One class is assigned exclusively to the forgetting client to represent unique forgetting knowledge, another class is shared across forgetting and retained clients to represent overlapping knowledge, and the remaining classes are distributed among retained clients. Table 5 summarizes the results.

All methods successfully achieve near-zero accuracy on the F-Acc metric, indicating effective removal of the client-specific unique data. However, a significant divergence appears in the handling of shared knowledge. Baselines such as FedEraser and FedOSD exhibit severe degradation in O-Acc, often dropping to near-zero performance. This indicates that these methods indiscriminately unlearn the entire class associated with the forgetting client, failing to distinguish between the specific samples to be forgotten and the general class knowledge held by the retained clients. In contrast, FUGAS preserves a significant portion of this shared knowledge, achieving O-Acc scores much closer to the Retraining baseline. Furthermore, FUGAS maintains superior performance on the unique retained knowledge, effectively matching the utility of the gold-standard model, demonstrating that our gradient projection mechanism successfully isolates the unlearning updates from the subspaces occupied by the retained tasks. These results confirm that FUGAS enables a more precise, instance-level erasure that protects the integrity of shared and retained representations.

6 Conclusion

In this work, we propose FUGAS, a unified gradient shaping framework that improves stability and retained performance without storing historical client updates. FUGAS stabilizes client-side unlearning through a bounded preference objective, and further preserves retained utility through a server-side compatibility projection. We show that FUGAS guarantees numerical stability and non-increasing retained empirical risk, and derive an excess risk bound that relates the final model to the retrained solution. Extensive experiments confirm that FUGAS achieves removal efficacy while maintaining competitive retained accuracy. FUGAS currently follows the standard trusted-server setting, but its projection step depends only on inner products between retained and forgetting updates. Since such inner-product computations are compatible with existing privacy-preserving protocols, an important future direction is a privacy-preserving extension of FUGAS so that the required projection statistics can be obtained without revealing raw individual gradients to the server.

Acknowledgements

This work was supported by National Key Research and Development Program of China under Grant no. 2022YFA1004102, National Natural Science Foundation of China under Grant no. 62136005, and Shenzhen fundamental research program JCYJ20250604144724032.

References

[1] G. Liu, X. Ma, Y. Yang, C. Wang, and J. Liu, "Federaser: Enabling efficient client-level data removal from federated learning models," in *IWQoS*, 2021.

[2] L. Wu, S. Guo, J. Wang, Z. Hong, J. Zhang, and Y. Ding, "Federated unlearning: Guarantee the right of clients to forget," *IEEE Network*, vol. 36, 2022.

[3] Z. Liu, Y. Jiang, J. Shen, M. Peng, K. Lam, X. Yuan, and X. Liu, "A survey on federated unlearning: Challenges, methods, and future directions," *ACM Comput. Surv.*, vol. 57, no. 1, pp. 2:1–2:38, 2025.

[4] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *AISTATS*, 2017.

[5] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, 2020.

[6] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proceedings of Machine Learning and Systems*, 2020, pp. 429–450.

[7] C. T. Dinh, N. Tran, and J. Nguyen, "Personalized federated learning with moreau envelopes," in *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2020, pp. 21 394–21 405.

[8] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. A. Bonawitz, Z. Charles *et al.*, "Advances and open problems in federated learning," *Found. Trends Mach. Learn.*, vol. 14, no. 1-2, pp. 1–210, 2021.

[9] A. Halimi, S. Kadhe, A. Rawat, and N. Baracaldo, "Federated unlearning: How to efficiently erase a client in fl?" *CoRR*, vol. abs/2207.05521, 2022.

[10] A. Dhasade, Y. Ding, S. Guo, A.-m. Kermarrec, M. De Vos, and L. Wu, "Quickdrop: Efficient federated unlearning by integrated dataset distillation," *arXiv preprint arXiv:2311.15603*, 2023.

[11] Z. Liu, H. Ye, Y. Jiang, J. Shen, J. Guo, I. Tjuawinata, and K.-Y. Lam, "Privacy-preserving federated unlearning with certified client removal," *IEEE*, vol. 20, pp. 3966–3978, 2025.

[12] G. Li, L. Shen, Y. Sun, Y. Hu, H. Hu, and D. Tao, "Subspace based federated unlearning," *CoRR*, vol. abs/2302.12448, 2023.

[13] M. Alam, H. Lamri, and M. Maniatakos, "Get rid of your trail: Remotely erasing backdoors in federated learning," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 12, pp. 6683–6698, 2024.

[14] N. Romandini, A. Mora, C. Mazzocca, R. Montanari, and P. Bellavista, "Federated unlearning: A survey on methods, design guidelines, and evaluation metrics," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 36, no. 7, pp. 11 697–11 717, 2025.

[15] Q. Li, Y. Diao, Q. Chen, and B. He, "Federated learning on non-iid data silos: An experimental study," in *ICDE*, 2022.

[16] Z. Lu, H. Pan, Y. Dai, X. Si, and Y. Zhang, "Federated learning with non-iid data: A survey," *IEEE Internet Things J.*, vol. 11, no. 11, pp. 19 188–19 209, 2024.

[17] M. Ye, X. Fang, B. Du, P. C. Yuen, and D. Tao, "Heterogeneous federated learning: State-of-the-art and research challenges," *ACM Comput. Surv.*, vol. 56, no. 3, pp. 79:1–79:44, 2024.

[18] L. Bourtoule, V. Chandrasekaran, C. A. Choquette-Choo, H. Jia, A. Travers, B. Zhang, D. Lie, and N. Papernot, "Machine unlearning," in *SP*, 2021.

[19] T. T. Nguyen, T. T. Huynh, P. L. Nguyen, A. W. Liew, H. Yin, and Q. V. H. Nguyen, "A survey of machine unlearning," *CoRR*, vol. abs/2209.02299, 2022.

[20] Y. Qu, X. Yuan, M. Ding, W. Ni, T. Rakotoarivelo, and D. Smith, "Learn to unlearn: A survey on machine unlearning," *arXiv preprint arXiv:2305.07512*, 2023.

[21] W. Wang, Z. Tian, C. Zhang, and S. Yu, "Machine unlearning: A comprehensive survey," *arXiv preprint arXiv:2405.07406*, 2024.

[22] V. Gupta, C. Jung, S. Neel, A. Roth, S. Sharifi-Malvajerdi, and C. Waites, "Adaptive machine unlearning," in *NeurIPS*, 2021.

[23] S. R. Kadhe, A. Halimi, A. Rawat, and N. Baracaldo, "Fairsisa: Ensemble post-processing to improve fairness of unlearning in llms," *CoRR*, vol. abs/2312.07420, 2023.

[24] N. G. Marchant, B. I. P. Rubinstein, and S. Alfeld, "Hard to forget: Poisoning attacks on certified machine unlearning," in *AAAI*, 2022.

[25] G. Wu, M. Hashemi, and C. Srinivasa, "PUMA: performance unchanged model augmentation for training data removal," in *AAAI*, 2022.

[26] S. Lin, X. Zhang, C. Chen, X. Chen, and W. Susilo, "ERM-KTP: knowledge-level machine unlearning via knowledge transfer," in *CVPR*, 2023.

[27] L. Han, N. Luo, H. Huang, J. Chen, and M. Hartley, "Towards independence criterion in machine unlearning of features and labels," *CoRR*, vol. abs/2403.08124, 2024.

[28] J. Foster, S. Schoepf, and A. Brintrup, "Fast machine unlearning without retraining through selective synaptic dampening," in *AAAI*, 2024.

[29] A. Thudi, G. Deza, V. Chandrasekaran, and N. Papernot, "Unrolling SGD: understanding factors influencing machine unlearning," in *EuroS&P*, 2022.

[30] A. K. Tarun, V. S. Chundawat, M. Mandal, and M. Kankanhalli, "Fast yet effective machine unlearning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, 2023.

[31] J. Chen, Z. Deng, K. Zheng, Y. Yan, S. Liu, P. Wu, P. Jiang, J. Liu, and X. Hu, "Safeeraser: Enhancing safety in multimodal large language models through multimodal machine unlearning," in *ACL (Findings)*, 2025.

[32] A. Peste, D. Alistarh, and C. H. Lampert, "Ssse: Efficiently erasing samples from trained machine learning models," *arXiv preprint arXiv:2107.03860*, 2021.

[33] A. Warnecke, L. Pirch, C. Wressnegger, and K. Rieck, "Machine unlearning of features and labels," in *NDSS*, 2023.

[34] X. Zhang, J. Wang, N. Cheng, Y. Sun, C. Zhang, and J. Xiao, "Machine unlearning methodology based on stochastic teacher network," in *International Conference on Advanced Data Mining and Applications*, 2023, pp. 250–261.

[35] L. Wang, T. Chen, W. Yuan, X. Zeng, K.-F. Wong, and H. Yin, "Kga: A general machine unlearning framework based on knowledge gap alignment," *arXiv preprint arXiv:2305.06535*, 2023.

[36] V. S. Chundawat, A. K. Tarun, M. Mandal, and M. Kankanhalli, "Zero-shot machine unlearning," *IEEE Transactions on Information Forensics and Security*, pp. 2345–2354, 2023.

[37] H. Kim, S. Lee, and S. S. Woo, "Layer attack unlearning: Fast and accurate machine unlearning via layer level attack and knowledge distillation," in *AAAI*, 2024.

[38] Y. Zhou, D. Zheng, Q. Mo, R. Lu, K. Lin, and W. Zheng, "Decoupled distillation to erase: A general unlearning method for any class-centric tasks," in *CVPR*, 2025.

[39] Y. Liu, L. Xu, X. Yuan, C. Wang, and B. Li, "The right to be forgotten in federated learning: An efficient realization with rapid retraining," in *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications*, 2022, pp. 1749–1758.

[40] N. Su and B. Li, "Asynchronous federated unlearning," in *INFOCOM*. IEEE, 2023, pp. 1–10.

[41] H. Jeong, S. Ma, and A. Houmansadr, "A survey on federated unlearning: Challenges and opportunities," *arXiv preprint arXiv:2403.02437*, 2024.

[42] X. Cao, J. Jia, Z. Zhang, and N. Z. Gong, "Fedrecover: Recovering from poisoning attacks in federated learning using historical information," in *2023 IEEE Symposium on Security and Privacy (SP)*, 2023, pp. 1366–1383.

[43] Z. Wang, D. M. Alghazzawi, L. Cheng, G. Liu, C. Wang, Z. Cheng, and Y. Yang, "Fedcsa: Boosting the convergence speed of federated unlearning under data heterogeneity," in *IEEE*, 2023, pp. 388–393.

[44] X. Guo, P. Wang, S. Qiu, W. Song, Q. Zhang, X. Wei, and D. Zhou, "FAST: adopting federated unlearning to eliminating malicious terminals at server side," *IEEE Trans. Netw. Sci. Eng.*, vol. 11, no. 2, pp. 2289–2302, 2024.

[45] L. Zhang, T. Zhu, H. Zhang, P. Xiong, and W. Zhou, "Fedrecovery: Differentially private machine unlearning for federated learning frameworks," *IEEE Trans. Inf. Forensics Secur.*, vol. 18, pp. 4732–4746, 2023.

[46] C. Wu, S. Zhu, and P. Mitra, "Federated unlearning with knowledge distillation," *CoRR*, vol. abs/2201.09441, 2022.

[47] W. Yuan, H. Yin, F. Wu, S. Zhang, T. He, and H. Wang, "Federated unlearning for on-device recommendation," in *WSDM*, 2023.

[48] J. Gong, O. Simeone, R. Kassab, and J. Kang, "Forget-svgd: Particle-based bayesian federated unlearning," *CoRR*, vol. abs/2111.12056, 2021.

[49] Y. H. Khalil, L. M. Brunswic, S. Lamghari, X. Li, M. Beitollahi, and X. Chen, "Not: Federated unlearning via weight negation," in *CVPR*. Computer Vision Foundation / IEEE, 2025, pp. 25 759–25 769.

[50] Z. Pan, Z. Wang, C. Li, K. Zheng, B. Wang, X. Tang, and J. Zhao, "Federated unlearning with gradient descent and conflict mitigation," in *AAAI*, 2025.

[51] C. Zhou, C. Pan, M. Li, and P. Wang, "Federated unlearning with fast recovery," *IEEE Trans. Mob. Comput.*, vol. 24, no. 10, pp. 9709–9725, 2025.

[52] C. Xie, K. Huang, P.-Y. Chen, and B. Li, "Dba: Distributed backdoor attacks against federated learning," in *International conference on learning representations*, 2019.

[53] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, 2020, pp. 2938–2948.

[54] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE Symposium on Security and Privacy (SP)*, 2017.

[55] L. Bai, H. Hu, Q. Ye, H. Li, L. Wang, and J. Xu, "Membership inference attacks and defenses in federated learning: A survey," *ACM Computing Surveys*, vol. 57, 2024.

A Derivation of the Dual Compatibility Projection

In this section, we provide the detailed derivation of the efficient dual optimization for the gradient projection mechanism presented in Eq. (2).

Recall the primal compatibility projection problem defined in Eq. (2):

$$\min_{\tilde{g}\in\mathbb{R}^d} \frac{1}{2}\|\tilde{g} - g_F^t\|_2^2 \quad \text{s.t.} \quad \mathbf{G}_R^t \tilde{g} \geq \mathbf{0}. \tag{6}$$

To solve this, we introduce the vector of Lagrange multipliers $\mathbf{v} \in \mathbb{R}^{|R|}$ with $\mathbf{v} \geq \mathbf{0}$. The Lagrangian function $\mathcal{L}(\tilde{g}, \mathbf{v})$ associated with the primal problem is given by

$$\mathcal{L}(\tilde{g}, \mathbf{v}) = \frac{1}{2}\|\tilde{g} - g_F^t\|_2^2 - \mathbf{v}^\top \mathbf{G}_R^t \tilde{g}. \tag{7}$$

Taking the derivative with respect to $\tilde{g}$ and setting it to zero gives

$$\nabla_{\tilde{g}}\mathcal{L} = \tilde{g} - g_F^t - {\mathbf{G}_R^t}^\top \mathbf{v} = 0 \implies \tilde{g}^* = g_F^t + {\mathbf{G}_R^t}^\top \mathbf{v}. \tag{8}$$

This confirms that the optimal update direction lies in the subspace spanned by the unlearning gradient and the retained gradients. To derive the dual objective function, we substitute the expression for $\tilde{g}^*$ back into the Lagrangian $\mathcal{L}(\tilde{g}, \mathbf{v})$:

$$\begin{aligned} \mathcal{L}(\mathbf{v}) &= \frac{1}{2}\|(g_F^t + {\mathbf{G}_R^t}^\top \mathbf{v}) - g_F^t\|_2^2 - \mathbf{v}^\top \mathbf{G}_R^t (g_F^t + {\mathbf{G}_R^t}^\top \mathbf{v}) \\ &= -\frac{1}{2}\mathbf{v}^\top (\mathbf{G}_R^t {\mathbf{G}_R^t}^\top)\mathbf{v} - \mathbf{v}^\top (\mathbf{G}_R^t g_F^t). \end{aligned} \tag{9}$$

The dual problem seeks to maximize this lower bound with respect to $\mathbf{v} \geq \mathbf{0}$:

$$\max_{\mathbf{v}\geq\mathbf{0}} \left(-\frac{1}{2}\mathbf{v}^\top \mathbf{P}\mathbf{v} - \mathbf{q}^\top \mathbf{v} \right), \tag{10}$$

where $\mathbf{P} = \mathbf{G}_R^t {\mathbf{G}_R^t}^\top$ and $\mathbf{q} = \mathbf{G}_R^t g_F^t$. Multiplying the objective by -1 transforms this maximization into the equivalent standard minimization Quadratic Programming (QP) form presented in Eq. (3):

$$\min_{\mathbf{v}\geq\mathbf{0}} \left(\frac{1}{2}\mathbf{v}^\top \mathbf{P}\mathbf{v} + \mathbf{q}^\top \mathbf{v} \right). \tag{11}$$

Once the optimal multipliers $\mathbf{v}^*$ are found using a QP solver, the primal solution is recovered via Eq. (8).

B Proofs of Propositions in the Main Text

In this section, we provide the complete proofs for the propositions presented in the main text regarding the stability, utility preservation, and approximation capabilities of FUGAS.

B.1 Proof of Proposition 1

Proof. Let $\mathbf{p} = f(x;\theta) \in \mathbb{R}^C$ denote the model output probability distribution, and let $\mathbf{p}^{\text{ref}}$ denote the fixed reference distribution cached from the pre-unlearning model. We define the scalar similarity score as $s = \mathbf{p}^\top \mathbf{p}^{\text{ref}}$ with $0 \leq s \leq 1$. The forgetting objective for a single sample can be written as

$$\mathcal{L}_{\text{forget}} = \log\left(1 + \exp\left(\frac{s}{\tau}\right)\right). \tag{12}$$

To find the gradient bound, we apply the chain rule:

$$\nabla_\theta \mathcal{L} = \frac{\partial \mathcal{L}}{\partial s} \cdot \nabla_\theta s. \tag{13}$$

First, calculating the derivative of the loss with respect to the similarity score s:

$$\frac{\partial \mathcal{L}}{\partial s} = \frac{1}{\tau} \cdot \frac{\exp\left(\frac{s}{\tau}\right)}{1 + \exp\left(\frac{s}{\tau}\right)}. \tag{14}$$

The fraction above lies in $(0, 1)$ for all s, therefore $0 < \frac{\partial \mathcal{L}}{\partial s} < \frac{1}{\tau}$.

Next, we compute the gradient with respect to the output distribution $\mathbf{p}$. Since $s(\theta) = \mathbf{p}^\top \mathbf{p}^{\text{ref}}$, we have $\nabla_{\mathbf{p}} s = \mathbf{p}^{\text{ref}}$ and by the chain rule,

$$\nabla_{\mathbf{p}} \mathcal{L} = \frac{\partial \mathcal{L}}{\partial s} \nabla_{\mathbf{p}} s = \frac{\partial \mathcal{L}}{\partial s} \mathbf{p}^{\text{ref}}. \tag{15}$$

Taking the ℓ_2 norm yields

$$\|\nabla_{\mathbf{p}} \mathcal{L}\|_2 \leq \left| \frac{\partial \mathcal{L}}{\partial s} \right| \|\mathbf{p}^{\text{ref}}\|_2 < \frac{1}{\tau} \|\mathbf{p}^{\text{ref}}\|_2. \tag{16}$$

Because $\mathbf{p}^{\text{ref}}$ is a probability vector, $\|\mathbf{p}^{\text{ref}}\|_2 \leq 1$, hence $\|\nabla_{\mathbf{p}} \mathcal{L}\|_2 < 1/\tau$.

Finally, we bound the parameter gradient. Using the chain rule again gives

$$\nabla_\theta \mathcal{L} = (\nabla_{\mathbf{p}} \mathcal{L})^\top \nabla_\theta \mathbf{p}, \tag{17}$$

so by submultiplicativity of operator norms,

$$\|\nabla_\theta \mathcal{L}\|_2 \leq \|\nabla_{\mathbf{p}} \mathcal{L}\|_2 \, \|\nabla_\theta \mathbf{p}\|_2. \tag{18}$$

Assumption 1 states that $\|\nabla_\theta \mathbf{p}\|_2 \leq L$, therefore $\|\nabla_\theta \ell\|_2 < L/\tau$ for every sample. Taking expectation over x preserves the bound, which yields $\|\nabla_\theta \mathcal{L}_{\text{forget}}\|_2 \leq L/\tau$. This completes the proof. □

B.2 Proof of Proposition 2

Proof. Let $\mathcal{L}_{\mathcal{R}}(\theta) = \sum_{k \in R} p_k \mathcal{L}(\mathcal{D}_k; \theta)$ denote the aggregate empirical risk over the retained clients. We analyze the change in this risk function after a single global update step $\theta^{t+1} = \theta^t - \eta_g \tilde{g}^t$, where $\eta_g > 0$ is the global learning rate and $\tilde{g}^t$ is the compatible update direction computed by FUGAS.

Consider the first-order Taylor expansion of the retained risk around the current parameters θ^t:

$$\mathcal{L}_{\mathcal{R}}(\theta^{t+1}) = \mathcal{L}_{\mathcal{R}}(\theta^t - \eta_g \tilde{g}^t) \approx \mathcal{L}_{\mathcal{R}}(\theta^t) - \eta_g \langle \nabla \mathcal{L}_{\mathcal{R}}(\theta^t), \tilde{g}^t \rangle. \tag{19}$$

The change in loss is governed by the first-order term:

$$\Delta \mathcal{L}_{\mathcal{R}} \approx -\eta_g \langle \nabla \mathcal{L}_{\mathcal{R}}(\theta^t), \tilde{g}^t \rangle. \tag{20}$$

To guarantee non-increasing risk, it just shows that the update direction $\tilde{g}^t$ has a non-negative inner product with the true gradient of the retained risk, $\langle \nabla \mathcal{L}_{\mathcal{R}}(\theta^t), \tilde{g}^t \rangle \geq 0$.

In the federated setting, the server aggregates gradients from retained clients to form the matrix $\mathbf{G}_R^t$, where each row corresponds to a client gradient g_k^t. The projection mechanism of FUGAS enforces the constraints:

$$\mathbf{G}_R^t \tilde{g}^t \geq \mathbf{0} \implies \langle g_k^t, \tilde{g}^t \rangle \geq 0, \quad \forall k \in R. \tag{21}$$

Since the weights p_k are non-negative, the inner product with the aggregate gradient satisfies:

$$\langle \nabla \mathcal{L}_{\mathcal{R}}(\theta^t), \tilde{g}^t \rangle \approx \sum_{k \in R} p_k \underbrace{\langle g_k^t, \tilde{g}^t \rangle}_{\geq 0} \geq 0. \tag{22}$$

Consequently, substituting this back into the Taylor expansion yields:

$$\Delta \mathcal{L}_{\mathcal{R}} \approx -\eta_g \underbrace{\langle \nabla \mathcal{L}_{\mathcal{R}}(\theta^t), \tilde{g}^t \rangle}_{\geq 0} \leq 0. \tag{23}$$

This confirms that under a first-order approximation, the gradient generated by FUGAS preserves utility on the retained clients. This completes the proof. □

B.3 Proof of Proposition 3

Proof. Let $\mathcal{L}_R(\theta)$ be the retained risk function, $\theta_R^* = \arg\min_\theta \mathcal{L}_R(\theta)$ be the gold-standard retrained model, and θ^0 be the pre-unlearning global model. We decompose the excess risk of the final unlearned model θ^T as

$$\begin{aligned}\mathcal{L}_R(\theta^T) - \mathcal{L}_R(\theta_R^*) =& \big(\mathcal{L}_R(\theta^0) - \mathcal{L}_R(\theta_R^*)\big) + \\ & \big(\mathcal{L}_R(\theta^T) - \mathcal{L}_R(\theta^0)\big).\end{aligned} \tag{24}$$

which includes the initial excess risk at the start of unlearning and the risk accumulation during the T unlearning rounds.

We first bound the initial drift term. The model θ^0 is obtained by minimizing the weighted training objective

$$\mathcal{L}(\theta) = (1-\alpha)\mathcal{L}_R(\theta) + \alpha\mathcal{L}_F(\theta), \tag{25}$$

where $\alpha = \sum_{k\in F} p_k$ is the weight of the forgetting clients. At optimality, $\nabla\mathcal{L}(\theta^0) = \mathbf{0}$, which implies

$$\nabla\mathcal{L}_R(\theta^0) = -\frac{\alpha}{1-\alpha}\nabla\mathcal{L}_F(\theta^0). \tag{26}$$

By μ-strong convexity of $\mathcal{L}_R$, we have for any θ that

$$\mathcal{L}_R(\theta) - \mathcal{L}_R(\theta_R^*) \le \frac{1}{2\mu}\|\nabla\mathcal{L}_R(\theta)\|_2^2. \tag{27}$$

Applying this inequality at $\theta = \theta^0$ and using Eq. (26) together with the gradient bound $\|\nabla\mathcal{L}_F(\theta^0)\|_2 \le G_F$ from Assumption 2 yields

$$\mathcal{L}_R(\theta^0) - \mathcal{L}_R(\theta_R^*) \le \frac{1}{2\mu}\left\|\frac{\alpha}{1-\alpha}\nabla\mathcal{L}_F(\theta^0)\right\|_2^2 \le \frac{\alpha^2 G_F^2}{2\mu(1-\alpha)^2}. \tag{28}$$

We next bound the algorithmic accumulation term. The server update is $\theta^{t+1} = \theta^t - \eta\tilde{g}^t$. By L-smoothness of $\mathcal{L}_R$,

$$\mathcal{L}_R(\theta^{t+1}) - \mathcal{L}_R(\theta^t) \le -\eta_g\langle\nabla\mathcal{L}_R(\theta^t), \tilde{g}^t\rangle + \frac{L\eta^2}{2}\|\tilde{g}^t\|_2^2. \tag{29}$$

While Proposition 2 guarantees a nonpositive first-order term $\langle\nabla\mathcal{L}_R(\theta^t), \tilde{g}^t\rangle$ in the current round under the first-order approximation. Using the update bound $\|\tilde{g}^t\|_2 \le G$ from Assumption 2, Eq. (29) gives

$$\mathcal{L}_R(\theta^{t+1}) - \mathcal{L}_R(\theta^t) \le \frac{L\eta_g^2 G^2}{2}. \tag{30}$$

Summing over $t = 0, \ldots, T-1$ yields

$$\mathcal{L}_R(\theta^T) - \mathcal{L}_R(\theta^0) \le \frac{TL\eta_g^2 G^2}{2}. \tag{31}$$

Combining Eq. (24), Eq. (28), and Eq. (31), we obtain the total excess risk bound as

$$\mathcal{L}_R(\theta^T) - \mathcal{L}_R(\theta_R^*) \le \frac{\alpha^2 G_F^2}{2\mu(1-\alpha)^2} + \frac{TL\eta_g^2 G^2}{2}. \tag{32}$$

This completes the proof. □

C Detailed Algorithmic Procedure

This section provides the comprehensive algorithmic workflow of FUGAS, supplementing the framework presented in Section 4. In this work, we focus on the federated unlearning setting where a subset of clients $F \subseteq K$ ($|F| \ge 1$) request unlearning and the remaining clients are $R = [K] \setminus F$.

Algorithm 1 Federated Unlearning with Gradient Adaptive Shaping (FUGAS)

Require: Pre-trained model θ^0, Unlearning rounds T, Client set K, Forgetting set F.
Server: Initialize $\theta^0 \leftarrow \theta_{\text{pre}}$.
Clients $k \in F$**:** Cache reference predictions $\mathbf{p}_k^{\text{ref}} \leftarrow f(\mathcal{D}_k; \theta^0)$.
for $t = 0$ **to** $T - 1$ **do**
  Server: Broadcast θ^t to all clients $k \in K$.
  for each client $k \in K$ **in parallel do**
    if $k \in F$ **then**
      Compute unlearning gradient $g_k^t = \nabla \mathcal{L}_{\text{forget}}(\mathcal{D}_k; \theta^t)$ via Eq. (1).
    else
      Compute standard gradient $g_k^t = \nabla \mathcal{L}_{\text{retain}}(\mathcal{D}_k; \theta^t)$.
    end if
    Send g_k^t to Server.
  end for
  Server Aggregation:
    Compute aggregated unlearning direction: $g_F^t \leftarrow \frac{1}{|F|} \sum_{k \in F} g_k^t$.
    Construct constraint matrix $\mathbf{G}_R^t$ from $\{g_k^t\}_{k \in R}$.
  Gradient Projection:
    Solve QP (Eq. (11)) to find Lagrange multipliers $\mathbf{v}^*$.
    Compute projected update: $\tilde{g}^t \leftarrow g_F^t + {\mathbf{G}_R^t}^\top \mathbf{v}^*$.
  Update: $\theta^{t+1} \leftarrow \theta^t - \tilde{g}^t$.
end for
return Unlearned global model θ^T.

Client-Side Local updates. At unlearning round t, the server broadcasts the current global model θ^t to all participating clients. Each retained client $k \in R$ performs standard local training on $\mathcal{D}_k$ and returns a model update. Each forgetting client $k \in F$ performs local unlearning using the bounded forgetting objective in Eq. (1) with the fixed negative reference $\mathbf{p}^{\text{ref}}$ computed from the pre-unlearning model θ^0 and returns its unlearning update.

Server-Side Global aggregation. FUGAS aggregates contributions of unlearning clients to form a unified unlearning direction. Specifically, after receiving local gradients g_k^t from all clients $k \in F$, the server computes the mean unlearning gradient:

$$g_F^t = \frac{1}{|F|} \sum_{k \in F} g_k^t. \tag{33}$$

This averaged vector represents the consensus direction for erasing the collective knowledge of the forgetting set.

For the retained clients $R = [K] \setminus F$, the server does not merely average their updates. Instead, to enforce the safety constraints, the server constructs the constraint matrix $\mathbf{G}_R^t \in \mathbb{R}^{|R| \times d}$, where each row corresponds to the individual gradient vector g_k^t of a retained client. This ensures that the final update $\tilde{g}^t$ satisfies $\langle \tilde{g}^t, g_k^t \rangle \geq 0$ for every retained client individually, providing a stricter and more robust guarantee than simply satisfying the constraint for the average retained gradient.

The server coordinates the process by broadcasting the current global model θ^t, collecting updates, solving the dual QP problem by Eq. (11) to obtain the optimal projection, and updating the global model. The complete procedure is summarized in Algorithm 1.

D Experiment

D.1 Detailed Experiment Setup

In this work, clients perform local optimization using SGD, batch size 32, and one local epoch per communication round. We first pre-train the global model for 500 rounds with a learning rate of 0.01 to obtain a stable initialization. We then conduct a 10-round unlearning phase in which the designated unlearning protocol is applied. We quantify the efficacy of unlearning using privacy and

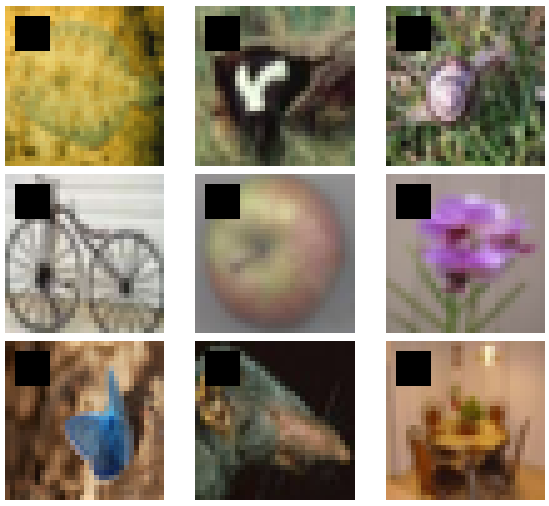


Figure 4: Examples of backdoor trigger images used in the unlearning evaluation. A fixed black patch is injected into training samples from the designated forgetting clients and the attack success rate measures whether the unlearned model still predicts triggered inputs as the target class.

security metrics while simultaneously monitoring model utility via test accuracy on retained clients. To measure the removal of influence, we employ the Attack Success Rate (ASR) metric derived from a backdoor attack in which a local pixel trigger is injected into all training samples of the forgetting clients. During pre-training, we inject a fixed trigger pattern into the designated unlearning client's training samples and relabel them to a target class. The attack success rate is the fraction of triggered test samples that are predicted as the target class, and effective unlearning drives this value close to that of a clean model. Furthermore, we evaluate the effectiveness of unlearning using a Membership Inference Attack (MIA) [54, 55] where a shadow model is used to train a binary classifier on prediction entropy and confidence scores to distinguish between training members and unseen non-members. A lower MIA accuracy indicates stronger privacy preservation and confirms that the unlearning process has successfully eliminated the unique statistical signals of the forgotten data, with values closer to 0.5 suggesting near-random inference. In addition, we report test accuracy on retained clients to measure utility preservation. We refer to Figure 4 for illustrative trigger examples.

D.2 Additional Experimental Results

Unlearning Evaluation of MIA. To further quantify the effectiveness afforded by our unlearning framework, we conduct MIA as described in Section 5.1. The goal of unlearning is to make the forgotten samples indistinguishable from non-training data, ideally driving the MIA accuracy down to 50%, representing the random guessing. Table 6 reports the MIA accuracy and retained classification accuracy for FUGAS and other baselines.

As expected, the pre-unlearning model exhibits very high MIA accuracy across all settings, indicating strong memorization signals about the training membership of the forgotten clients. While baselines like PGD and FedEraser reduce this leakage, they often do so at the cost of retained utility. However, FUGAS consistently achieves the lowest MIA accuracy while maintaining the highest retained accuracy among all unlearning methods. This confirms that our preference-based objective effectively eliminates the unique statistical signals of the forgetting data without destroying the generalizable features required for the remaining tasks.

Impact of Post-Training Recovery. To assess the long-term stability of the unlearned model, we conduct an additional 10 rounds of post-training on the retained clients immediately following the unlearning phase. This phase serves two purposes including evaluating whether the model can recover any lost utility and verifying that the forgotten knowledge does not spontaneously re-emerge. Table 7 summarizes the performance after this recovery phase.

Table 6: Unlearning efficacy under membership inference attacks and retained accuracy on CIFAR-10 and CIFAR-100 benchmarks. All values are percentages, with accuracy reported as mean ± standard deviation. $|F|$ denotes the number of forgetting clients. The highest retained accuracy among unlearning methods is bolded.

Method	CIFAR-10 non-IID		CIFAR-10 IID		CIFAR-100 non-IID		CIFAR-100 IID	
	MIA → 50	Acc ↑	MIA → 50	Acc ↑	MIA → 50	Acc ↑	MIA → 50	Acc ↑
$\|F\| = 1$								
Pre-Unlearning θ^0	97.9	$76.2_{\pm 0.5}$	97.5	$80.5_{\pm 0.9}$	97.8	$50.6_{\pm 0.2}$	97.8	$54.4_{\pm 1.1}$
Retraining	74.8	$78.9_{\pm 0.1}$	74.8	$81.2_{\pm 0.0}$	74.2	$53.7_{\pm 0.4}$	74.8	$52.8_{\pm 0.6}$
PGD	74.6	$57.8_{\pm 0.2}$	80.8	$70.5_{\pm 1.8}$	75.1	$32.0_{\pm 0.7}$	78.4	$37.8_{\pm 2.6}$
FedEraser	72.1	$30.4_{\pm 0.9}$	76.2	$67.3_{\pm 1.3}$	77.5	$30.7_{\pm 0.9}$	76.6	$35.4_{\pm 0.8}$
FedOSD	74.9	$63.7_{\pm 0.2}$	75.1	$73.9_{\pm 3.7}$	75.9	$36.6_{\pm 1.3}$	77.2	$42.5_{\pm 0.1}$
FUGAS (Ours)	70.5	$\mathbf{70.9}_{\pm 0.7}$	74.3	$\mathbf{79.8}_{\pm 2.7}$	70.1	$\mathbf{49.5}_{\pm 0.5}$	74.1	$\mathbf{51.5}_{\pm 0.7}$
$\|F\| = 2$								
Pre-Unlearning θ^0	97.9	$76.2_{\pm 0.5}$	97.5	$80.5_{\pm 0.9}$	97.8	$50.6_{\pm 0.2}$	97.8	$54.4_{\pm 1.1}$
Retraining	75.1	$80.9_{\pm 0.1}$	75.6	$83.8_{\pm 0.0}$	73.9	$52.1_{\pm 0.2}$	75.3	$53.2_{\pm 0.8}$
PGD	76.1	$\mathbf{74.4}_{\pm 6.3}$	82.4	$78.0_{\pm 0.4}$	74.3	$35.6_{\pm 0.7}$	79.5	$38.4_{\pm 2.4}$
FedEraser	73.2	$28.9_{\pm 2.6}$	77.3	$67.8_{\pm 1.6}$	75.8	$32.2_{\pm 0.7}$	76.7	$37.9_{\pm 1.2}$
FedOSD	75.4	$65.2_{\pm 0.7}$	76.7	$75.4_{\pm 0.2}$	75.6	$34.8_{\pm 0.5}$	79.0	$43.4_{\pm 2.7}$
FUGAS (Ours)	70.4	$74.3_{\pm 0.2}$	75.5	$\mathbf{80.9}_{\pm 2.6}$	72.7	$\mathbf{49.9}_{\pm 0.4}$	74.6	$\mathbf{50.2}_{\pm 0.1}$

Table 7: Unlearning efficacy and retained accuracy after post-training on CIFAR-10 and CIFAR-100 benchmarks. All values are percentages, with accuracy reported as mean ± standard deviation. $|F|$ denotes the number of forgetting clients. The highest retained accuracy among unlearning methods is bolded.

Method	CIFAR-10 non-IID		CIFAR-10 IID		CIFAR-100 non-IID		CIFAR-100 IID	
	ASR ↓	Acc ↑	ASR ↓	Acc ↑	ASR ↓	Acc ↑	ASR ↓	Acc ↑
$\|F\| = 1$								
Pre-Unlearning θ^0	99.8	$62.5_{\pm 1.0}$	99.7	$81.2_{\pm 1.6}$	97.5	$41.4_{\pm 5.0}$	99.2	$49.3_{\pm 1.0}$
Retraining	1.3	$71.2_{\pm 0.7}$	1.7	$80.9_{\pm 0.3}$	0.0	$50.1_{\pm 0.1}$	0.0	$52.9_{\pm 0.5}$
FedEraser	2.2	$31.3_{\pm 0.3}$	4.8	$75.1_{\pm 0.2}$	0.2	$35.3_{\pm 0.3}$	0.2	$45.6_{\pm 0.2}$
FedOSD	3.4	$71.6_{\pm 0.7}$	4.5	$79.3_{\pm 0.1}$	0.1	$49.7_{\pm 0.1}$	0.0	$50.6_{\pm 0.0}$
FUGAS (Ours)	3.6	$\mathbf{73.4}_{\pm 0.2}$	4.5	$\mathbf{79.8}_{\pm 1.2}$	0.4	$\mathbf{50.3}_{\pm 0.1}$	0.0	$\mathbf{51.1}_{\pm 0.1}$
$\|F\| = 2$								
Pre-Unlearning θ^0	99.9	$41.9_{\pm 0.9}$	99.9	$72.8_{\pm 0.7}$	99.8	$28.6_{\pm 1.2}$	99.9	$38.9_{\pm 1.3}$
Retraining	0.8	$79.8_{\pm 0.1}$	0.2	$77.6_{\pm 0.1}$	0.0	$51.3_{\pm 0.6}$	0.0	$48.7_{\pm 0.2}$
FedEraser	3.3	$34.7_{\pm 0.0}$	4.9	$74.2_{\pm 0.0}$	0.4	$43.2_{\pm 0.2}$	3.3	$33.8_{\pm 0.0}$
FedOSD	9.7	$60.8_{\pm 1.1}$	16.3	$77.3_{\pm 0.1}$	0.8	$48.4_{\pm 0.9}$	0.1	$45.1_{\pm 0.2}$
FUGAS (Ours)	12.4	$\mathbf{65.9}_{\pm 0.3}$	2.1	$\mathbf{77.6}_{\pm 0.2}$	0.1	$\mathbf{49.9}_{\pm 0.2}$	0.0	$\mathbf{48.0}_{\pm 0.1}$

While methods like FedEraser and FedOSD show some utility recovery, they often exhibit a trade-off where ASR begins to creep back up. In contrast, FUGAS demonstrates superior stability even after this recovery phase. Notably, FUGAS achieves comparable or higher accuracy with lower ASR, indicating that our method does not overly forget in a way that requires aggressive repair. This stability suggests that FUGAS effectively integrates the unlearning and retaining objectives in a single coherent process, reducing the dependency on computationally expensive post-hoc fixes. More importantly, this indicates that our gradient projection mechanism creates a lasting geometric separation between the forgetting and retaining subspaces, ensuring that subsequent standard training on retained data does not inadvertently "undo" the unlearning operation.

Table 8: Unlearning efficacy and retained accuracy on CIFAR-10 and CIFAR-100 benchmarks under a high-skew Dirichlet split with $\alpha = 0.02$ and a pathological non-IID split. All values are percentages, with accuracy reported as mean $\pm$ standard deviation. The highest retained accuracy among approximate unlearning methods is bolded.

Method	Dirichlet non-IID with $\alpha = 0.02$				Pathological non-IID			
	CIFAR-10		CIFAR-100		CIFAR-10		CIFAR-100	
	ASR ↓	Acc ↑	ASR ↓	Acc ↑	ASR ↓	Acc ↑	ASR ↓	Acc ↑
Pre-Unlearning θ^0	98.1	$61.7_{\pm 0.1}$	99.7	$47.2_{\pm 0.3}$	95.2	$88.2_{\pm 0.1}$	99.7	$46.9_{\pm 0.3}$
Retraining	0.1	$74.4_{\pm 0.5}$	0.0	$57.6_{\pm 2.3}$	1.3	$87.3_{\pm 0.0}$	0.0	$65.2_{\pm 0.4}$
PGD	0.5	$31.3_{\pm 0.9}$	0.0	$17.9_{\pm 1.1}$	0.5	$34.3_{\pm 0.1}$	2.1	$24.1_{\pm 0.5}$
FedBU	0.0	$23.3_{\pm 0.1}$	0.0	$48.0_{\pm 0.1}$	0.0	$54.6_{\pm 0.0}$	0.0	$38.2_{\pm 0.9}$
FedEraser	0.0	$28.6_{\pm 0.5}$	0.0	$21.6_{\pm 1.2}$	0.0	$34.8_{\pm 0.1}$	0.0	$25.2_{\pm 0.2}$
FedOSD	0.1	$57.6_{\pm 0.3}$	0.0	$48.9_{\pm 0.1}$	0.1	$72.0_{\pm 0.1}$	0.0	$54.6_{\pm 0.5}$
FUGAS (Ours)	0.0	$\mathbf{66.2}_{\pm 0.0}$	0.0	$\mathbf{56.1}_{\pm 0.2}$	0.4	$\mathbf{83.3}_{\pm 0.0}$	0.0	$\mathbf{62.4}_{\pm 0.1}$

Table 9: Unlearning efficacy and retained accuracy on CIFAR-10 and CIFAR-100 benchmarks with $K = 30$ clients. All values are percentages, with accuracy reported as mean $\pm$ standard deviation. The highest retained accuracy among unlearning methods is bolded.

Method	CIFAR-10 non-IID		CIFAR-10 IID		CIFAR-100 non-IID		CIFAR-100 IID	
	ASR ↓	Acc ↑	ASR ↓	Acc ↑	ASR ↓	Acc ↑	ASR ↓	Acc ↑
Pre-Unlearning θ^0	82.6	$51.8_{\pm 0.1}$	87.2	$64.7_{\pm 1.2}$	92.6	$40.9_{\pm 1.2}$	87.8	$45.8_{\pm 1.3}$
Retraining	0.7	$60.7_{\pm 0.6}$	0.4	$75.5_{\pm 0.0}$	0.0	$46.8_{\pm 0.3}$	0.0	$48.9_{\pm 0.1}$
PGD	3.4	$39.2_{\pm 0.5}$	1.5	$59.3_{\pm 0.9}$	0.0	$35.6_{\pm 0.1}$	0.8	$40.1_{\pm 0.3}$
FedBU	0.0	$33.7_{\pm 0.2}$	0.0	$61.7_{\pm 2.2}$	3.1	$37.3_{\pm 0.9}$	0.4	$43.7_{\pm 0.2}$
FedEraser	0.2	$23.1_{\pm 0.2}$	0.0	$53.9_{\pm 0.2}$	0.0	$31.9_{\pm 0.2}$	0.2	$38.2_{\pm 0.1}$
FedOSD	0.0	$46.0_{\pm 0.1}$	0.3	$66.9_{\pm 0.1}$	0.0	$39.2_{\pm 0.1}$	0.0	$46.8_{\pm 1.2}$
FUGAS (Ours)	0.1	$\mathbf{54.6}_{\pm 0.2}$	0.1	$\mathbf{72.9}_{\pm 0.2}$	0.0	$\mathbf{44.1}_{\pm 0.0}$	0.0	$\mathbf{48.2}_{\pm 0.0}$

Robustness under Extreme Heterogeneity. To further test the robustness of our gradient projection mechanism, we evaluate FUGAS under two extreme heterogeneity scenarios including high-skew Dirichlet distribution and Pathological non-IID setting. In high-skew Dirichlet setting, we set $\alpha = 0.02$ to create a highly imbalanced partition where clients possess data from very few classes. In pathological non-IID setting, we restrict each client to select exactly 2 categories from 10 categories in CIFAR-10 or 20 out of 100 in CIFAR-100, thereby creating maximal gradient divergence.

Under baselines like PGD and FedEraser, model utility collapses catastrophically due to the intense destructive interference between the unlearning and retaining gradients. FedOSD, which relies on strict orthogonal projections, also struggles to maintain utility in the pathological setting. In sharp contrast, FUGAS maintains high retained accuracy while successfully suppressing the backdoor ASR to near zero. This result provides strong empirical evidence that our dynamic gradient projection mechanism is uniquely capable of identifying and preserving compatible update directions even when the local data distributions are disjoint, validating the robustness of FUGAS under highly challenging and realistic heterogeneity.

Scalability with More Clients. We expand our experimental evaluation to a larger federation consisting of $K = 30$ clients compared to $K = 10$ in the main results. Table 9 presents the unlearning efficacy and retained utility in this expanded setting.

Although several baselines can suppress ASR, they often suffer noticeable utility degradation, especially in the non-IID setting where aggregation becomes more conflicted. Despite the increased complexity of the constraint landscape, FUGAS continues to deliver superior performance. It consistently achieves near-zero ASR while preserving high classification accuracy, significantly outperforming baselines like PGD and FedEraser, which suffer from notable utility degradation. This result confirms that our flexible acute-angle constraint is sufficiently permissive to identify valid

Table 10: Unlearning efficacy and retained accuracy of FUGAS on CIFAR-10 and CIFAR-100 benchmarks under different retained client join rates during the unlearning phase. All values are percentages, with accuracy reported as mean ± standard deviation. The highest retained accuracy is bolded.

Join Rate	CIFAR-10 non-IID		CIFAR-10 IID		CIFAR-100 non-IID		CIFAR-100 IID	
	ASR↓	Acc↑	ASR↓	Acc↑	ASR↓	Acc↑	ASR↓	Acc↑
20%	0.0	**67.4**$_{\pm 0.3}$	0.2	75.2$_{\pm 0.2}$	0.0	50.6$_{\pm 0.1}$	0.0	51.0$_{\pm 0.2}$
50%	0.0	65.9$_{\pm 0.0}$	0.0	76.8$_{\pm 0.4}$	0.0	50.7$_{\pm 1.2}$	0.0	**52.3**$_{\pm 0.9}$
100%	0.1	67.2$_{\pm 0.0}$	0.0	**78.7**$_{\pm 0.0}$	0.0	**50.9**$_{\pm 0.2}$	0.0	51.3$_{\pm 0.1}$

Table 11: Unlearning efficacy and retained accuracy on Tiny-ImageNet with ResNet-50, 100 clients, and a participation rate of 20%. All values are percentages, with accuracy reported as mean ± standard deviation. The highest retained accuracy among unlearning methods is bolded.

Method	non-IID		IID	
	ASR↓	Acc↑	ASR↓	Acc↑
Pre-Unlearning θ^0	100.0	57.5$_{\pm 0.3}$	100.0	61.0$_{\pm 0.1}$
Retraining	0.0	58.1$_{\pm 0.5}$	0.2	62.5$_{\pm 0.7}$
FedEraser	0.0	41.8$_{\pm 0.9}$	0.0	48.6$_{\pm 0.2}$
FedBU	17.8	44.0$_{\pm 0.5}$	14.6	47.1$_{\pm 0.1}$
FedOSD	0.0	53.8$_{\pm 0.2}$	0.5	55.6$_{\pm 0.5}$
NoT	0.7	52.5$_{\pm 0.3}$	2.4	54.4$_{\pm 0.0}$
FUGAS (Ours)	0.0	**57.2**$_{\pm 0.4}$	0.0	**59.7**$_{\pm 0.3}$

update directions even in more populated and constrained environments, validating the scalability of FUGAS for larger-scale deployments.

Robustness with Varying Client Join Rates. We further analyze the impact of client join rates during the unlearning phase, ranging from 20% to 100%. Table 10 presents the unlearning efficacy and retained utility under different participation levels. FUGAS achieves negligible ASR without compromising classification accuracy, regardless of participation fluctuations. Such resilience to dynamic client availability emphasizes its effectiveness in balancing unlearning efficiency with utility preservation, affirming its viability for real-world federated deployments.

Scalability to Larger Models and Sparse Participation. To evaluate FUGAS in a more challenging federated setting, we conduct experiments on Tiny-ImageNet with 200 classes using ResNet-50 and 100 clients, with 20% of clients participating in each round. As shown in Table 11, FedBU fails to remove the backdoor in this setting, while FedEraser, FedOSD, and NoT reduce ASR at a noticeable cost in retained accuracy. FUGAS drives ASR to zero and keeps retained accuracy close to that of the pre-unlearning model, achieving the highest accuracy among unlearning methods under both partitions. These results demonstrate the effectiveness of FUGAS with a larger model and more clients under partial client participation.

Generalization to Text Classification. To examine whether FUGAS extends beyond vision, we evaluate it on the AGNews news classification benchmark with a TextCNN. Following the backdoor unlearning setting, we construct the trigger by replacing the first five tokens of each input with a fixed token and assigning the target label. As shown in Table 12, FedEraser leaves a residual backdoor under the non-IID partition, and FedBU and FedOSD remove the backdoor but suffer substantial utility degradation. FUGAS removes the backdoor under both partitions and attains clearly higher retained accuracy than all baselines, with a wider margin over the strongest baseline than on the image benchmarks. These results indicate that the gradient shaping of FUGAS is not tied to a particular input modality.

Table 12: Unlearning efficacy and retained accuracy on AGNews with TextCNN. All values are percentages, with accuracy reported as mean ± standard deviation. The highest retained accuracy among approximate unlearning methods is bolded.

Method	non-IID		IID	
	ASR↓	Acc↑	ASR↓	Acc↑
Pre-Unlearning θ^0	100.0	$80.6_{\pm 0.1}$	98.3	$84.7_{\pm 0.2}$
Retraining	0.0	$86.1_{\pm 0.3}$	0.0	$89.9_{\pm 0.0}$
FedEraser	19.2	$54.9_{\pm 0.7}$	0.9	$55.3_{\pm 1.2}$
FedBU	0.0	$62.4_{\pm 0.5}$	0.0	$64.6_{\pm 1.0}$
FedOSD	0.0	$65.0_{\pm 0.7}$	0.0	$70.9_{\pm 0.9}$
FUGAS (Ours)	0.0	$\mathbf{74.7}_{\pm 1.9}$	0.4	$\mathbf{79.3}_{\pm 1.3}$

Table 13: Sensitivity analysis of the temperature parameter τ in the preference-based forgetting objective on CIFAR-10 and CIFAR-100 benchmarks. All values are percentages, with accuracy reported as mean ± standard deviation. The highest retained accuracy for each setting is bolded.

τ	CIFAR-10 non-IID		CIFAR-10 IID		CIFAR-100 non-IID		CIFAR-100 IID	
	ASR↓	Acc↑	ASR↓	Acc↑	ASR↓	Acc↑	ASR↓	Acc↑
1	0.0	$66.9_{\pm 0.1}$	0.0	$75.7_{\pm 0.0}$	0.0	$48.7_{\pm 0.2}$	0.0	$50.8_{\pm 0.1}$
2	0.1	$67.8_{\pm 0.1}$	0.4	$78.4_{\pm 0.0}$	0.0	$49.1_{\pm 0.1}$	0.1	$51.0_{\pm 0.2}$
3	0.1	$67.2_{\pm 0.0}$	0.0	$78.7_{\pm 0.0}$	0.0	$\mathbf{50.9}_{\pm 0.2}$	0.0	$\mathbf{51.3}_{\pm 0.1}$
4	0.1	$67.0_{\pm 0.2}$	5.8	$\mathbf{79.1}_{\pm 0.1}$	0.0	$49.2_{\pm 0.1}$	0.0	$51.2_{\pm 0.0}$
5	0.1	$\mathbf{68.2}_{\pm 0.1}$	7.6	$78.9_{\pm 0.0}$	0.0	$49.0_{\pm 0.0}$	0.0	$51.2_{\pm 0.0}$

D.3 Sensitivity Analysis of Temperature

The forgetting loss in Eq. (1) introduces a critical temperature hyperparameter τ, which controls the intensity of the divergence penalty. To investigate the sensitivity of FUGAS to this parameter, we conduct an ablation study by varying τ across a range of values $\{1, \ldots, 5\}$ and evaluating the trade-off between unlearning efficacy (ASR) and utility preservation (Accuracy), while keeping all other training settings identical to the main experiments.

Concretely, τ rescales the similarity term between the current output distribution $\mathbf{p}$ and the cached pre-unlearning reference $\mathbf{p}^{\text{ref}}$. A smaller τ creates a sharper, more aggressive penalty for outputs that remain similar to the forgotten predictions, potentially accelerating unlearning but risking utility degradation. Conversely, a larger τ results in a softer, more moderate penalty, which may preserve utility better but could slow down the erasure process.

Table 13 presents the performance across different settings. The results demonstrate that FUGAS exhibits remarkable robustness to hyperparameter choice. Specifically, the method maintains an excellent balance of near-zero ASR and high retained accuracy across a wide range of $\tau \in [1, 5]$. While extremely high values slightly improve retained accuracy on the IID setting, they begin to show a marginal increase in ASR in challenging scenarios, indicating a trade-off where the unlearning signal becomes too weak. Based on these results, we use $\tau = 3$ as the default choice in our experiments. Overall, this analysis confirms that FUGAS does not require fine-grained tuning of τ to achieve state-of-the-art performance, highlighting its practicality for real-world deployment.

D.4 Robustness under Unconverged Initial Conditions

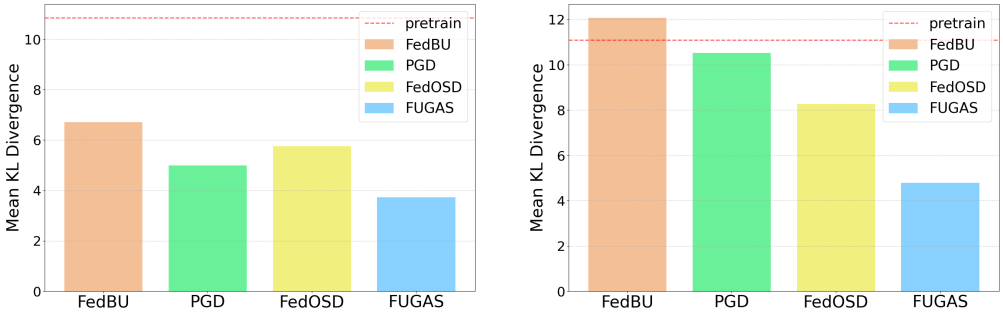
We assume that the pre-unlearning global model is already converged, while unlearning requests may arrive at any stage of the model lifecycle, including when the model is under-fitted in practical federated systems. To empirically examine the robustness of FUGAS when this assumption is violated, we conduct additional experiments where the pre-unlearning model is unconverged. Instead of the standard 500 pre-training rounds used in the main experiment to obtain a stable and well-converged initialization, we initialize the unlearning phase after only 150 rounds of federated training. At this stage, the under-trained pre-unlearning model exhibits noticeably lower utility than the fully trained

Table 14: Unlearning efficacy and retained accuracy on CIFAR-10 and CIFAR-100 benchmarks starting from an unconverged global model. The pre-unlearning model θ^0 is obtained after 150 rather than 500 rounds of federated pre-training. All values are percentages, with accuracy reported as mean $\pm$ standard deviation. The highest retained accuracy among unlearning methods is bolded.

Method	CIFAR-10 non-IID		CIFAR-10 IID		CIFAR-100 non-IID		CIFAR-100 IID	
	ASR ↓	Acc ↑	ASR ↓	Acc ↑	ASR ↓	Acc ↑	ASR ↓	Acc ↑
Pre-Unlearning θ^0	94.7	$54.8_{\pm 3.1}$	99.7	$77.6_{\pm 1.2}$	71.9	$35.3_{\pm 1.0}$	79.8	$42.6_{\pm 1.0}$
Retraining	0.0	$69.2_{\pm 0.3}$	2.0	$77.3_{\pm 0.2}$	0.0	$43.5_{\pm 0.8}$	0.0	$47.4_{\pm 0.3}$
PGD	0.0	$37.9_{\pm 1.1}$	0.1	$65.3_{\pm 3.2}$	0.0	$36.2_{\pm 0.1}$	0.0	$39.4_{\pm 0.1}$
FedBU	0.0	$21.8_{\pm 1.3}$	0.0	$52.9_{\pm 0.0}$	0.3	$19.7_{\pm 1.3}$	1.2	$36.0_{\pm 0.1}$
FedEraser	0.2	$32.2_{\pm 0.0}$	0.1	$40.2_{\pm 0.7}$	0.1	$24.8_{\pm 0.2}$	0.2	$29.8_{\pm 0.2}$
FedOSD	0.0	$44.2_{\pm 0.5}$	0.0	$71.0_{\pm 0.0}$	0.0	$40.3_{\pm 0.8}$	0.0	$45.8_{\pm 0.1}$
FUGAS (Ours)	0.1	$\mathbf{67.2}_{\pm 0.0}$	1.0	$\mathbf{76.0}_{\pm 0.9}$	0.2	$\mathbf{42.5}_{\pm 0.0}$	0.1	$\mathbf{48.2}_{\pm 0.1}$

counterpart. Table 14 reports the unlearning efficacy (ASR) and retained accuracy starting from this under-trained θ_0.

Table 14 shows that PGD and FedBU can drive ASR to near zero but incur substantial utility loss, indicating that aggressive loss-maximization style forgetting remains brittle when the starting point is not well-optimized. FedEraser also exhibits limited retained accuracy, suggesting that reconstruction from historical updates is not sufficient to stabilize utility in this under-trained regime. FedOSD provides a better balance, yet it still falls short of FUGAS in retained accuracy. Despite this more challenging initialization, FUGAS still achieves a strong the trade-off between forgetting and utility. It consistently suppresses ASR to near-zero while maintaining the highest retained accuracy among memory-efficient unlearning methods across all settings. These findings demonstrate that FUGAS is uniquely capable of turning the unlearning phase into a constructive process for the retained tasks, making it highly suitable for dynamic FL environments where models are continuously evolving.



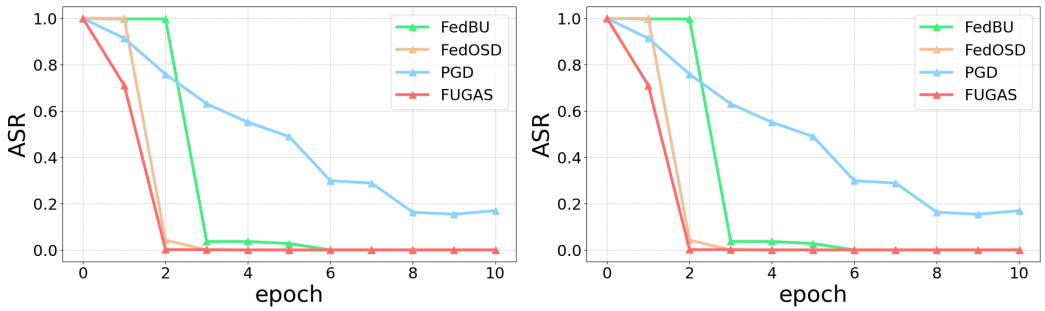
(a) CIFAR-10 (b) CIFAR-100

Figure 5: Mean KL divergence between the output distributions of unlearned models and the gold-standard Retraining model on the forgetting dataset. The dashed "pretrain" line indicates the pre-unlearning baseline, serving as upper bound.

D.5 Distributional Alignment With Retraining

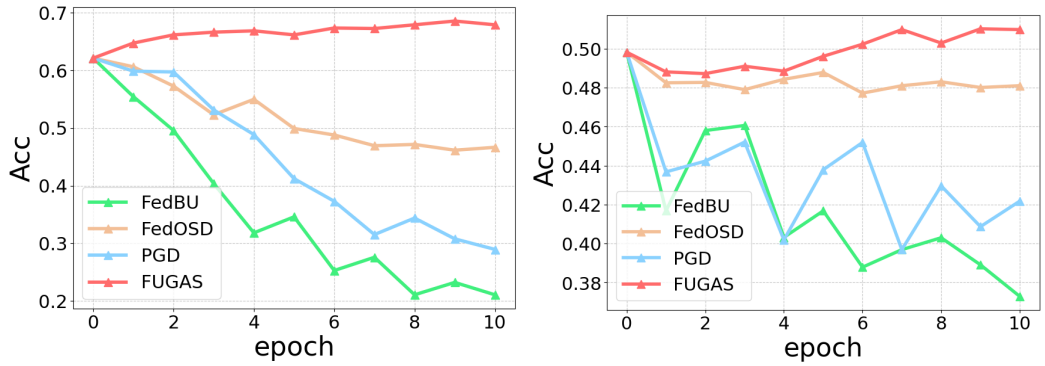
To provide a more granular evaluation of unlearning, we analyze the statistical similarity between the unlearned models and the gold-standard Retrained model. Specifically, we compute the Kullback-Leibler (KL) divergence between the prediction distributions produced by different unlearning methods and the gold-standard Retraining model. A lower KL divergence indicates that the unlearned model has successfully approximated the predictive behavior of a model trained from scratch, rather than merely suppressing specific class activations. We visualize the result in Figure 5.

The dashed red line represents the pre-unlearning model as the baseline upper bound for divergence, reflecting that the backdoor-induced behavior substantially differs from the clean retrained solution. As shown, baseline methods such as FedBU and PGD exhibit relatively high KL divergence, suggesting that while they may reduce ASR, their internal representations and output distributions remain distinct from the optimal retraining. FedOSD shows improvement but still maintains a noticeable gap. In contrast, FUGAS achieves the lowest KL divergence across benchmarks. This superior distributional alignment confirms that FUGAS enables the model to traverse the optimization landscape toward a state that is statistically indistinguishable from the gold standard, validating our theoretical claims of approximation.



(a) CIFAR-10 Non-IID (b) CIFAR-100 Non-IID

Figure 6: Unlearning efficacy on poison data. We evaluate the forgetting capability of FUGAS and baselines by measuring the ASR on the poison data for CIFAR-10 and CIFAR-100 under Non-IID setting.



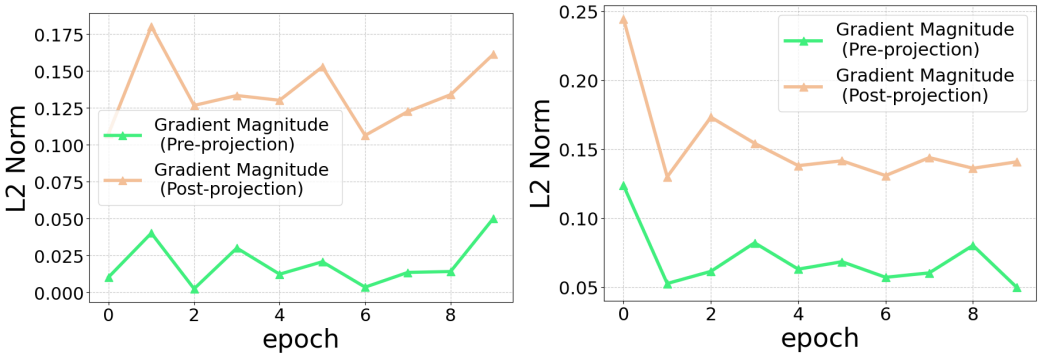
(a) CIFAR-10 Non-IID (b) CIFAR-100 Non-IID

Figure 7: Retained utility in remaining clients. We track the accuracy evolution of FUGAS and baselines throughout the unlearning process on CIFAR-10 and CIFAR-100 under Non-IID setting.

D.6 Dynamic Behavior Throughout the Unlearning Process

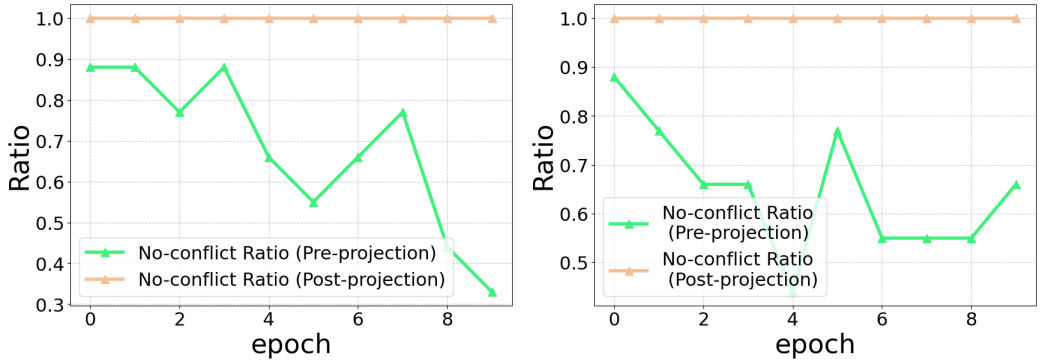
This subsection examines the dynamic behavior of FUGAS across the unlearning phase. We first investigate the unlearning efficacy on poison data. We then analyze the model utility preservation on remaining clients through accuracy evolution. Finally, we inspect the internal mechanics of the gradient projection module, validating that the QP solver maintains viable gradient norms and achieves complete conflict elimination throughout the optimization.

Unlearning Efficacy on Poison Data. The ASR is used to measure whether the model has successfully forgotten the poison data. As shown in Figure 6, FUGAS achieves a rapid and complete drop in ASR to nearly 0 within the first few epochs. In contrast, baseline methods like PGD exhibit slow decay, failing to ensure complete forgetting. This confirms that FUGAS can thoroughly eliminate the influence of the poison data.



(a) CIFAR-10 IID (b) CIFAR-10 Non-IID

Figure 8: Feasibility of the gradient projection step in federated unlearning. We examine the ℓ_2 norm of the projected gradient to verify that it remains non-zero, ensuring the QP solver converges to a valid solution.



(a) CIFAR-10 IID (b) CIFAR-10 Non-IID

Figure 9: Gradient conflict analysis in federated unlearning. We evaluate the conflict-free ratio of gradient updates between retained and unlearning clients before and after projection on CIFAR-10.

Retained Utility in Remaining Clients. The Accuracy reflects the model's performance on the remaining clients shown in Figure 7. While competitors like FedBU and PGD suffer catastrophic drops in accuracy, FUGAS maintains stable and high accuracy throughout the unlearning process. It effectively balances the trade-off between forgetting and retaining utility. In summary, FUGAS stands out by simultaneously achieving near-zero ASR and high retained accuracy, proving its effectiveness and robustness in federated unlearning scenarios.

Feasibility of Gradient Projection. To validate the practicality of the proposed QP solver, we present empirical evidence along two dimensions. First, we compare the ℓ_2 norm of updated gradients before and after projection. As shown in Figure 8, the post-projection gradient norms remain strictly non-zero, confirming that the QP solver does not collapse gradients to zero and thus preserves meaningful parameter updates. Second, we measure the ratio of conflict-free gradients across both stages shown in Figure 9. While partial gradient conflicts are observed prior to projection, the QP mapping successfully eliminates all such conflicts, achieving 100% post-projection conflict-free rate. These findings demonstrate that the projection mechanism admits a broad feasible region, effectively reconciling conflict removal with stable gradient propagation.

D.7 Unlearning under Differential Privacy

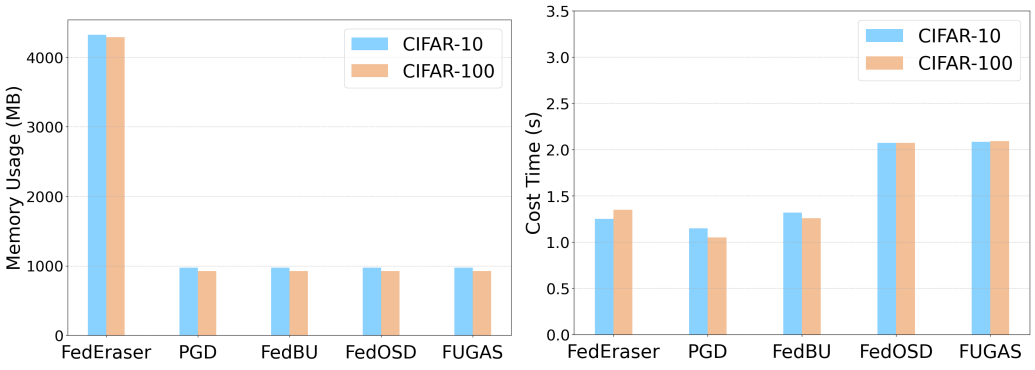
Section 5.4 hides individual retained gradients from the server, yet the aggregate itself may still reveal information about client data. Therefore, we examine whether FUGAS remains effective when retained updates are privatized with the Gaussian mechanism. In each unlearning round, every

Table 15: Unlearning efficacy and retained accuracy with differentially private retained-client updates on CIFAR-10 and CIFAR-100 benchmarks. All values are percentages, with accuracy reported as mean ± standard deviation. The highest retained accuracy among unlearning methods is bolded.

Method	CIFAR-10 non-IID		CIFAR-10 IID		CIFAR-100 non-IID		CIFAR-100 IID	
	ASR ↓	Acc ↑	ASR ↓	Acc ↑	ASR ↓	Acc ↑	ASR ↓	Acc ↑
Pre-Unlearning θ^0	99.8	$62.5_{\pm 1.0}$	99.7	$81.2_{\pm 1.6}$	97.5	$41.4_{\pm 5.0}$	99.2	$49.3_{\pm 1.0}$
FedOSD	0.2	$46.8_{\pm 0.8}$	0.0	$62.4_{\pm 0.1}$	0.3	$28.8_{\pm 0.5}$	0.0	$38.7_{\pm 0.3}$
FUGAS (Ours)	0.7	$\mathbf{52.6}_{\pm 0.0}$	0.0	$\mathbf{68.4}_{\pm 0.1}$	0.1	$\mathbf{34.3}_{\pm 0.1}$	0.0	$\mathbf{40.5}_{\pm 0.7}$

retained client $k \in R$ clips its update to an ℓ_2 normand adds Gaussian noise before uploading it. We compose the mechanism over the $T = 10$ unlearning rounds with a privacy budget of $\epsilon = 14.65$ at $\delta = 10^{-5}$. We evaluate the unlearning performance under noisy retained-client updates under both IID and non-IID partitions..

Table 15 compares FUGAS with FedOSD, which also projects with retained gradients and is exposed to the same noise. FUGAS maintains low ASR across all four settings and consistently achieves higher retained accuracy than FedOSD. Although DP noise reduces retained utility, these results demonstrate that FUGAS remains effective under the evaluated privacy setting. Stronger privacy guarantees remains directions for future work.



(a) Peak memory usage

(b) Average per round time

Figure 10: Computational overhead of federated unlearning. We compare memory and time costs of FUGAS and baselines on CIFAR-10 and CIFAR-100. (a) Peak memory overhead (MB) during unlearning. (b) Average per-round wall-clock time (s) during unlearning.

D.8 Computational and Memory Efficiency Analysis

We further evaluate the efficiency of federated unlearning by measuring both memory overhead and runtime per unlearning round on CIFAR-10 and CIFAR-100, as shown in Figure 10. To reflect practical FL deployments where clients run in parallel, we report the per-round wall-clock time as the maximum local runtime among participating clients plus the server aggregation time. Memory usage is reported as the peak memory during the unlearning stage.

As shown in Figure 10(a), FedEraser incurs substantially higher memory overhead due to storing and replaying historical updates for reconstruction. In contrast, FUGAS matches the low memory of memory-efficient baselines, since it does not require storing client histories or auxiliary datasets beyond the standard model and per-round updates. This property is consistent with our design goal of memory-efficient federated unlearning. Figure 10(b) shows that FUGAS incurs a marginal increase in computation time, primarily from the server-side dual QP solving used in gradient projection. However, when contextualized with the utility results in Table 1, FUGAS invests this marginal computational cost to achieve utility comparable to the computationally exorbitant Retraining baseline. FUGAS introduces minimal additional memory and computational overhead. The reference cache

adds only 0.177 MB and 1.752 MB of extra memory on CIFAR-10 and CIFAR-100, accounting for merely 0.02% and 0.19% of the peak memory usage, respectively. The server-side QP solving takes 0.030 s and 0.034 s, constituting only 1.44% and 1.63% of the per-round communication time. These results demonstrate that the algorithm achieves efficient selective forgetting with negligible overhead. Overall, FUGAS delivers effective unlearning and remains high accuracy on retained clients, and it does so with a memory usage comparable to the most lightweight baselines and only minor runtime overhead.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The main contributions, including the FUGAS framework, theoretical stability guarantees, and empirical results, are clearly stated in the abstract and Section 1. These claims are supported by technical derivations in Section 4 and experiments in Section 5.

   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.

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discuss the computational overhead of the dual QP solver and potential extensions to large-scale foundation models in Section 6 and Section D.8.

   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.

3. **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: All assumptions for our stability and convergence analysis are explicitly stated in Section 4.3 at Assumptions 1 and 2. Complete formal proofs for Propositions 1, 2, and 3 are provided in Appendix B.

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.

4. **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: We specify all experimental settings, including dataset splits such as IID and Non-IID, architectures such as ResNet-18, and hyperparameters such as temperature $\tau = 3$, unlearning rounds $T = 10$ in Section 5.1 and Appendix D.1.

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.

5. **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: Anonymized code and detailed environment setup instructions are provided in the supplemental material. We use standard publicly available datasets such as CIFAR-10/100.

   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.

6. **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: Full details regarding pre-training rounds, unlearning rounds, batch size, and the grid search range for learning rates are provided in Section 5.1 and Appendix D.1.

   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.

7. **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 results in Tables 1-15 are reported as the mean $\pm$ standard deviation across three random seeds, and the factors of variability are stated in Section 5.1.

   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.

8. **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: We provide a detailed analysis of peak memory usage (MB) and average per-round wall-clock time in Figure 10 and Section D.8.

   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).

9. **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: We have reviewed the NeurIPS Code of Ethics. Our work on federated unlearning is designed to protect user privacy and the right to be forgotten, adhering to ethical standards.

   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).

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: The positive impact on data privacy is discussed in Section 1. We also analyze potential security implications through our attack-based evaluations in Section 5.

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).

11. **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: [N/A]

    Justification: Our work focuses on a general unlearning algorithm and does not release high-risk pre-trained models or scraped datasets.

    Guidelines: (Skipped in output)

    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.

12. **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: All datasets and baseline codes used are cited in the References section.

    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.

13. **New assets**

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

    Answer: [Yes]

    Justification: The FUGAS algorithm is fully documented via pseudocode in Algorithm 1 and implementation details in the appendix.

    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.

14. **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: Our research does not involve crowdsourcing or 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.

15. **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: Our research does not involve 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.

16. **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: The core methodology development does not involve LLMs.

    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.