NoT: Federated Unlearning via Weight Negation

Yasser H. Khalil[1]* Leo Brunswic[1]* Soufiane Lamghari[1] Xu Li[2] Mahdi Beitollahi[1] Xi Chen[1]
[1]Huawei Noah's Ark Lab, Montreal, Canada [2]Huawei Technologies Canada Inc., Ottawa, Canada

Abstract

Federated unlearning (FU) aims to remove a participant's data contributions from a trained federated learning (FL) model, ensuring privacy and regulatory compliance. Traditional FU methods often depend on auxiliary storage on either the client or server side or require direct access to the data targeted for removal—a dependency that may not be feasible if the data is no longer available. To overcome these limitations, we propose ***NoT****, a novel and efficient FU algorithm based on weight negation (multiplying by -1), which circumvents the need for additional storage and access to the target data. We argue that effective and efficient unlearning can be achieved by perturbing model parameters away from the set of optimal parameters, yet being well-positioned for quick re-optimization. This technique, though seemingly contradictory, is theoretically grounded: we prove that the weight negation perturbation effectively disrupts inter-layer co-adaptation, inducing unlearning while preserving an approximate optimality property, thereby enabling rapid recovery. Experimental results across three datasets and three model architectures demonstrate that NoT significantly outperforms existing baselines in unlearning efficacy as well as in communication and computational efficiency.*

1. Introduction

Federated learning (FL) enables decentralized machine learning across distributed devices, allowing models to be trained collaboratively without sharing raw data, thus enhancing privacy and security [32, 43, 70]. However, growing concerns over privacy and data security have emphasized the need for unlearning techniques to meet evolving regulatory standards [40, 65]. Federated unlearning (FU) addresses this by enabling removal of individual data contributions from trained FL models [39, 51]. This capability is essential for privacy preservation and compliance with regulations such as GDPR [12], which mandates the "right to be forgotten." FU is also critical when data is outdated, compromised, or subject to data poisoning attacks [61].

While various FU techniques have been proposed [21, 38, 63, 71], they face significant challenges in FL environments. For instance, exact unlearning methods such as retraining from scratch guarantee thorough data removal but are impractical due to high communication and computational demands. Other FU approaches require additional storage for model updates, which may be infeasible and could pose additional security risks. Additionally, many existing methods depend on having access to the target data, which may no longer be available or permissible for use.

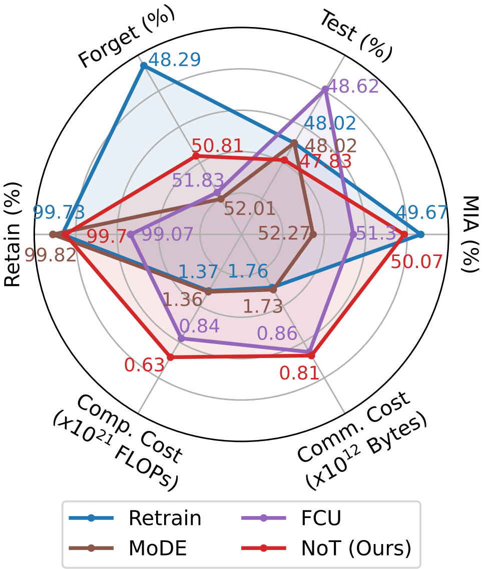

Figure 1. **Performance comparison of NoT with baselines** using ViT-B/16 on Caltech-101 in a 10-client setup, where one client requests unlearning. The ideal federated unlearning algorithm should closely approximate the performance of the "gold standard" (Retrain) across key accuracy metrics: *retain*, *forget*, *test*, and *MIA*, while minimizing communication and computation overhead. As illustrated, NoT's performance closely matches that of Retrain across all metrics with minimal added costs, underscoring NoT's efficacy and efficiency in federated unlearning. Experimental details and further comparisons can be found in Section 6.

To address these limitations, we propose NoT, a novel and efficient FU algorithm based on weight negation, requiring neither auxiliary storage nor access to the target data. NoT operates by negating (multiplying by -1) the pa-

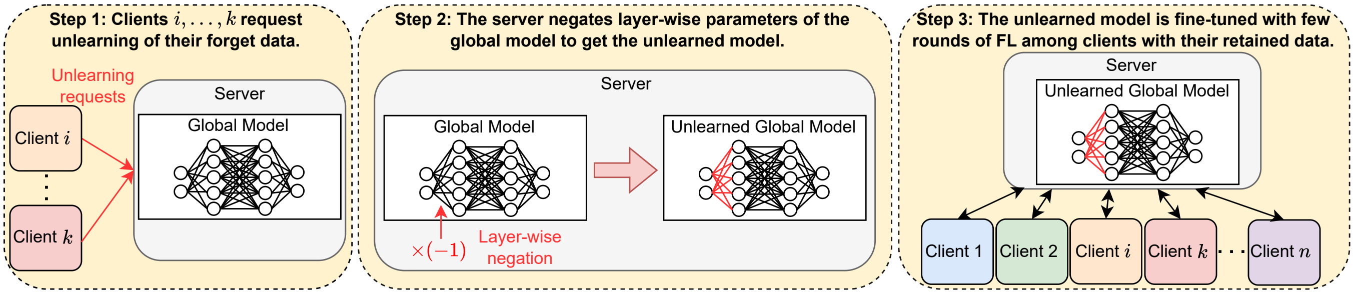
Step 1: Clients $i, \ldots, k$ request unlearning of their forget data.
Unlearning requests
Client i
Client k
Server
Global Model
Step 2: The server negates layer-wise parameters of the global model to get the unlearned model.
Server
Global Model
Unlearned Global Model
$\times(-1)$
Layer-wise negation
Step 3: The unlearned model is fine-tuned with few rounds of FL among clients with their retained data.
Server
Unlearned Global Model
Client 1
Client 2
Client i
Client k
Client n

Figure 2. **NoT overview.** Upon receiving unlearning requests from target clients, the server initiates the unlearning process by applying layer-wise parameter negation to the global model. This negation disrupts inter-layer co-adaptation, effectively inducing unlearning. Subsequent fine-tuning rounds restore essential knowledge. If a client wishes to forget all its data (i.e., client-wise forgetting), it does not participate in fine-tuning. Conversely, if a client wants partial data forgetting (i.e., class-wise or instance-wise forgetting), it fine-tunes the global model using its retained data.

rameters of specific layers in the global model, as depicted in Figure 2. The name **NoT** reflects the Boolean "NOT" operation. Negation breaks inter-layer co-adaptation (*i.e.*, dependencies between network parameters [24, 53], see Appendix 9), resulting generically in high loss, which effectively "forgets" the targeted data. Fine-tuning on retained data subsequently allows for recovery of essential knowledge. Together, these two phases induce significant parameter changes that remove knowledge of the data to be unlearned. To formalize this approach, we incorporate NoT within a novel theoretical unlearning framework, establishing that effective and efficient unlearning can be achieved by *perturbing model parameters away from the optimal set of parameters, yet being well-positioned for re-optimization*. Our empirical results confirm that NoT achieves effective unlearning while minimizing communication and computation compared to state-of-the-art methods. As illustrated in Figure 1, NoT achieves performance close to the "gold standard" Retrain method across several accuracy metrics, while significantly reducing communication and computation costs. Moreover, since NoT does not rely on access to the forgot data, it naturally supports client-wise, class-wise, and instance-wise forgetting. Our main **contributions** are summarized as follows:

- We propose NoT, an efficient FU algorithm leveraging weight negation, operating without requiring additional storage or access to target data.
- We present a theoretical framework describing how unlearning is achieved through weight perturbation and fine-tuning. We provide an effective bound controlling the unlearning via fine-tuning, introduce the notion of layer-wise optimality which enables fast-recovery, and prove that weight negation conforms to this framework. Further, we empirically validate our theoretical predictions.
- We conduct an extensive experimental evaluation of NoT, benchmarking it against seven FU methods across three datasets and three model architectures. Our evaluation includes experiments addressing backdoor attacks, tests in centralized settings with eight baselines, and an ablation study.

2. Related Work

Federated Unlearning (FU). Two main approaches dominate FU [8, 16, 21, 38, 46, 49, 63, 69, 71]: ❶ **Storing historical updates**: These methods save prior model updates for later use in unlearning specific data. For instance, FedEraser [38] retrieves the global model state before a client joins the federation, sharing it with the remaining clients to remove the target client's influence. FUKD [63] subtracts the target client's updates and uses knowledge distillation to restore model performance. However, these approaches pose privacy risks due to potential model update leakage [49], face storage limitations, and often require unlabeled data, which may not always be available. Despite these challenges, they do not require the target client's participation during unlearning. ❷ **Gradient modification**: These methods alter model gradients during training to suppress the impact of target data. For example, PGD [21] reverses the learning process for the target client, constraining the update within an ℓ_2-norm ball around a reference model, which is then fine-tuned by the remaining clients. MoDE [71] uses a randomly initialized degradation model for unlearning, while FCU [8] simulates a model that has never seen the forgotten data by applying contrastive loss and preserving low-frequency components of the global model. These methods, however, often require the target client's involvement and impose high computational costs. In contrast, *NoT eliminates the need for extra storage or access to the target data.*

Unlearning via Weight Perturbation. Several works focus on unlearning through weight modification. Golatkar *et al.* [18] proposes adding Gaussian noise, computed using Fisher information, to disrupt weights for unlearning. In a follow-up, Golatkar *et al*. [19] employ the Neural Tangent Kernel (NTK) to address the null space of model weights. Tarun *et al*. [58] introduces an error-maximizing noise matrix trained on a pretrained model to corrupt weights associated with specific target classes. A more targeted approach, SSD [14], dampens parameters deemed sensitive to forgotten data using Fisher information. Most weight perturbation methods that focus on class-based forgetting, however, struggle with random data forgetting and are computationally expensive due to Fisher information calculations. *To our knowledge, no prior work explores unlearning through weight negation.* In our results, we compare NoT's negation approach with other perturbation methods.

3. Preliminaries

We define a *model* as a parameterized family $\mathcal{N}^\theta : \mathbb{R}^{d_{\text{in}}} \to \mathbb{R}^{d_{\text{out}}}$, where $\theta \in \Theta \subset \mathbb{R}^d$, such that the map $(x, \theta) \mapsto \mathcal{N}^\theta(x)$ is continuous and piecewise twice continuously differentiable. We focus on models provided by neural networks, the parameters tensor may then be written as $\theta = (\theta_\ell)_{\ell \in \mathscr{L}}$ where $\mathscr{L}$ is the set of layers. We assume a dataset-dependent non-negative loss function, $\mathcal{L}_D(\mathcal{N}) := \mathbb{E}_{(x,y)\sim D} L(\mathcal{N}(x), y)$, and that the model is typically trained via gradient descent to minimize this loss. For simplicity, we denote the loss as $\mathcal{L}_D(\theta)$ instead of $\mathcal{L}_D(\mathcal{N}^\theta)$, when the context clearly refers to the model $\mathcal{N}^\theta$.

In this paper, we consider an FL system with n clients, where each client $k \in \mathcal{P} = \{1, \ldots, n\}$ has local training data D^k. Clients collaboratively train a global model until convergence $\mathcal{N}^{\theta^*}$ using a standard FL algorithm, such as FedAvg. After training, each client k may request the server to unlearn a subset of its data $D_u^k \subseteq D^k$, referred to as the target or forget data, while $D_r^k := D^k \setminus D_u^k$ represents its retained data. The client requesting unlearning is called the target client. A straightforward solution to unlearn $D_u = \bigcup_k D_u^k$ is to retrain the model from scratch using distributed $D_r = \bigcup_k D_r^k$, resulting in a retrained model. However, this approach is computationally and communication-intensive. *The challenge is to efficiently find an unlearned model that closely approximates the performance of the retrain model.*

4. Unlearning Framework

Given a trained model $\mathcal{N}^{\theta^*}$, our objective is to efficiently unlearn data without retraining from scratch. We propose a two-step approach: first, **perturbing** model parameters to obtain perturbed parameters θ', followed by **fine-tuning** using gradient descent starting at $\theta^0 = \theta'$ to minimize $\mathcal{L}_{D_r}$, resulting in the unlearned model. The underlying intuition behind this approach is that perturbing the model induces not only immediate unlearning but also large gradients for the subsequent fine-tuning phase, hence substantial alteration of the model's internal configuration and promoting further unlearning. However, the perturbation should be designed to avoid an excessive fine-tuning phase.

> Informally, we seek a perturbation that is:
> ✦ **(C1)** ***Strong***: *significantly pushes model parameters away from optimal configurations.*
> ✦ **(C2)** ***Resilient***: *enables fast re-optimization.*

4.1. The Need for a Strong Perturbation

To motivate condition **C1**, we introduce the concept of **loss gap** as a measure for unlearning.

Definition 1 (Loss gap)**.** *Let $\mathcal{N}^\theta$ be a model, and let (D_r, D_u) be a pair of datasets. The* ***loss gap*** *is defined as:*

$$\delta(\theta) := |\mathcal{L}_{D_r}(\theta) - \mathcal{L}_{D_u}(\theta)|. \qquad (1)$$

In unlearning, the goal is to increase the loss gap by a target amount, ensuring that $\mathcal{L}_{D_r}$ is minimized while $\mathcal{L}_{D_u}$ is not. The rationale behind **C1** is supported by the following theorem:

Theorem 1. *Let $\mathcal{N}^\theta$ be a model, and let (D_r, D_u) be a pair of datasets. Given an initial parameter set $\theta^0 \in \Theta$, assume $\mathcal{N}^{\theta^0}$ is trained using Stochastic Gradient Langevin Descent*[1] *to minimize $\mathcal{L}_{D_r}$ starting from θ^0. The parameter evolution is given by: $d\theta^t = -\nabla_{\theta^t}\mathcal{L}dt + \Sigma(\theta^t, t) \cdot dW$. At any training time $t \geq 0$, the following holds:*[2]

$$t \geq \frac{\mathbb{E}(\delta(\theta^t) - \delta(\theta^0))^2}{L^2\left[|\mathcal{L}_{D_r}(\theta^0) - \mathbb{E}\mathcal{L}_{D_r}(\theta^t)| + A\right]}, \qquad (2)$$

with $L := \sup_{\theta_1 \neq \theta_2} \frac{|\delta(\theta_1) - \delta(\theta_2)|}{\|\theta_1 - \theta_2\|}$ and A depends explicitly on Σ and the Hessian of $\mathcal{L}_{D_r}$ along the solution, and $A = 0$ when $\Sigma \equiv 0$ (see Appendix 12.1 for proof and details).

Here, the left side of the inequality represents the time t required for unlearning via gradient descent. The numerator on the right indicates the unlearning target, while the denominator depends on datasets characteristics, loss function and training stochasticity. This theorem suggests that unlearning may be *slow* if some conditions are not met. For instance, natural forgetting ($\theta^0 = \theta' = \theta^*$) leads to small L due to minor statistical differences between D_r and D_u, and a negligible $|\mathcal{L}_{D_r}(\theta^0) - \mathbb{E}\mathcal{L}_{D_r}(\theta^t)|$ as $\mathcal{N}^{\theta^0} = \mathcal{N}^{\theta^*}$ is already converged. Therefore, Theorem 1 applied to the following fine-tuning on D_r predicts extended unlearning time, hence slow natural forgetting. See Appendix 13.1 for quantitative estimation. For fast unlearning, factors such as a large Hessian spectrum, high loss, or high stochasticity during descent are essential, hence a **strong** perturbation that increases loss is beneficial.

[1] SGLD may be seen as an approximation of SGD, see [56].

[2] Expectations taken over the randomness of the stochastic process θ^t.

4.2. What is a Resilient Perturbation?

There are various strong perturbations (*e.g.*, randomization of θ), but they may require intensive fine-tuning. Therefore, condition **C2** seeks to ensure that the perturbed set of parameters θ' is in a good optimization state, allowing efficient recovery during fine-tuning. A good optimization state could be defined *a posteriori* as one where $t^*_\varepsilon := \min\{t \mid \mathcal{L}(\theta^t) - \mathcal{L}(\theta^*) \leq \varepsilon\}$ is small, relative to a baseline (such as retraining from scratch). However, we favor *a priori* properties of the neural network at training time $t = 0$ yielding (statistically) a small t^*_ε. Existing evidence suggests that certain properties ease the optimization of a model:

ⓐ **Jacobian Control**: typically analyzed through the spectrum of $(\nabla_x \mathcal{N}^\theta)^T(\nabla_x \mathcal{N}^\theta)$ or $(\nabla_\theta \mathcal{N}^\theta)^T(\nabla_\theta \mathcal{N}^\theta)$, it controls gradient back-propagation. Accumulation of the eigenvalues of the former around 1 is coined dynamical isometry [25, 54]. The spectrum of the latter relates to the spectrum of the Fisher Information Matrix (FIM) [1, 48]. Dynamical isometry is expected in networks with residual connections [3, 45, 57] or well-chosen initializations [4, 5, 47, 64]. It controls the FIM spectrum [29, 37] which in turn impacts gradient dynamics via control over the stochasticity of gradient descent [56] and isospectrality to the Neural Tangent Kernel [27] in the mean field limit.

ⓑ **Model Pretraining** is empirically shown to accelerate fine-tuning [10, 11, 23, 66, 68]. While this is not fully theoretically accounted for, a pretrained model is expected to require less learning of low-level features, reducing the fine-tuning search space compared to training from scratch.

In our framework, denoting $J(\theta, x) := \nabla_\theta \mathcal{N}^\theta\big|_x$ and X a random variable sampled from D_r, we translate Jacobian control as a control over distributions of random matrices $J(\theta', X)$ and $J(\theta^*, X)$. If, for example, the Wasserstein distance $\mathcal{W}(J(\theta^*, X), J(\theta', X))$ is sufficiently small [60], gradients should behave comparably during fine-tuning of $\mathcal{N}^{\theta'}$ relative to the original training which yielded $\mathcal{N}^{\theta^*}$. A perturbation has *Jacobian control* if it satisfies a bound on this Wasserstein distance. Also, since the pre-perturbation model $\mathcal{N}^{\theta^*}$ is converged, it is effectively pretrained on the dataset $D_r \cup D_u$. *Layer-wise optimality* formalizes the preservation of part of the effective pretraining and ensures a smaller dimension of the fine-tuning search space.

Definition 2 (Layer-Wise Optimality)**.** *A model $\mathcal{N}^\theta$ is layer-wise optimal (LWO) if, for every chosen layer ℓ, freezing ℓ, randomizing all other layers $\ell' \neq \ell$, and fine-tuning still yield an acceptable optimum.*

If $\mathcal{N}^{\theta'}$ remains LWO when $\mathcal{N}^\theta$ is LWO, then the perturbation $\theta \mapsto \theta'$ is layer-wise optimality preserving (LWOP).

Together, Jacobian control and layer-wise optimality guarantee accelerated fine-tuning. Thus, a perturbation that is LWOP and has Jacobian control is considered **Resilient**.

> *To conclude Section 4, achieving both conditions requires perturbing to maximize loss (**C1**) while controlling the model's Jacobian (**C2a**) and preserving layer-wise optimality (**C2b**).*

5. NoT - The Unlearning Algorithm

In this section, we present the NoT algorithm and discuss its role as a federated unlearning solution. First, we outline the algorithm's design, process, and advantages. Second, we position NoT within our theoretical framework of perturbation and fine-tuning for unlearning. Finally, we discuss the selection of layers for negation.

5.1. Algorithm Overview

When an unlearning request is received, the server initiates the unlearning process by **negating** the parameters of specified layers $\mathscr{L}_{\text{neg}}$ in the converged global model $\mathcal{N}^{\theta^*}$. This produces a perturbed model $\mathcal{N}^{\theta'}$ with parameters θ' as follows:

$$\theta' := (-\theta^*_\ell)_{\ell \in \mathscr{L}_{\text{neg}}} \oplus (\theta^*_\ell)_{\ell \in \mathscr{L} \setminus \mathscr{L}_{\text{neg}}}. \tag{3}$$

NoT then fine-tunes $\mathcal{N}^{\theta'}$ on the retained data D_r, resulting in a final model that excludes target data contributions while preserving essential knowledge. Algorithm 1 in Appendix 10 details our proposed method via weight negation. A PyTorch implementation is included in Appendix 11.

NoT presents several a priori advantages: ❶ Negation is *computationally negligible*. ❷ The target client only needs to signal an unlearning request, thus *communication cost is minimal*. ❸ *No additional storage required* on the client or server side. ❹ *No access to D_u* is needed, allowing target clients to delete it immediately after requesting unlearning. While there are some costs incurred during fine-tuning, they are relatively low as we will show empirically in our experiments.

5.2. Negation as a Strong and Resilient Perturbation

The NoT algorithm follows a "perturb then fine-tune" approach. Now, we theoretically ensure that NoT's negation-based perturbation meets the conditions **(C1)** and **(C2)** for effective unlearning. Let $\mathcal{N}^\theta$ be a neural network model and let X be a random vector following a dataset distribution $\mathcal{D}$, we denote by Y_ℓ the pre-nonlinearity activations of layer $\ell \in \mathscr{L}$ given input X and define $Y_- := \bigoplus_{\ell \in \mathscr{L}_{\text{neg}}} Y_\ell$.

✦ **(C1)** Under mild assumptions, weight negation is the strongest perturbation as it maximizes changes in the activations of the perturbed layers:

Theorem 2. *Denote $\sigma(x) := \max(x, 0)$ and let Y'_ℓ be the output of layer ℓ for some perturbation of ℓ. Assume $\mathbb{E}\left|\|\sigma(Y_\ell)\|^2 - \|\sigma(-Y_\ell)\|^2\right| \leq \varepsilon$ and*

$\mathbb{E}\left|\|\sigma(Y_\ell)\|^2 - \|\sigma(Y'_\ell)\|^2\right| \leq \varepsilon$, *then:*

$$\mathbb{E}\|\sigma(Y_\ell) - \sigma(-Y_\ell)\|^2 \geq \mathbb{E}\|\sigma(Y_\ell) - \sigma(Y'_\ell)\|^2 - 2\varepsilon. \quad (4)$$

While the loss may occasionally remain unchanged (*e.g.*, if the next layer is zero), the distinct distribution of (X, Y_-) from $(X, -Y_-)$ implies that **C1** is generally satisfied. See Appendix 12.2 for proof and additional discussion.

✦ **(C2)** The following Theorems ensure conditions **C2a** and **C2b**, ensuring resilience for NoT.

Theorem 3. *Assume $\mathcal{N}^\theta$ is a feedforward*[3] *neural network of layer poset $(\mathscr{L}, \leq)$ ordered by the computational graph. Let $J_\ell^\theta := \nabla_{\theta_\ell}\mathcal{N}^\theta(X)$ with $X \sim \mathcal{D}$ for layer $\ell \in \mathscr{L}$. Then:*

$$\begin{aligned}
\mathcal{W}\left(J_\ell^{\theta^*}; J_\ell^{\theta'}\right) &\leq A_\ell \operatorname{TV}(Y_-; -Y_-), & \forall \ell > \mathscr{L}_{\text{neg}} \\
\mathcal{W}\left(\epsilon J_\ell^{\theta^*}; J_\ell^{\theta'}\right) &\leq A_\ell \operatorname{TV}((X, Y_-); (X, -Y_-)), & \forall \ell \leq \mathscr{L}_{\text{neg}}
\end{aligned} \quad (5)$$

where $\epsilon = (-1)^{\ell \notin \mathscr{L}_{\text{neg}}}$, $\mathcal{W}$ and TV are the Wasserstein and total variation distances, respectively, and $(A_\ell)_{\ell \in \mathscr{L}}$ are positive values. See Appendix 12.3 for the proof, technical assumptions, and details on A_ℓ.

Theorem 4. *The negation perturbation is LWOP if $\mathscr{L}_{\text{neg}}$ is an antichain of the poset $\mathscr{L}$ containing no maximal element, and each $\ell \in \mathscr{L}_{\text{neg}}$ is activated by sigmoid-like, odd, or even functions (*e.g.*, $\mathbf{1}_{>0}$, $\tanh$, $\sin$, x^2). See Appendix 12.4 for details and proof.*

In the wide network limit, we expect $\operatorname{TV}(Y_\ell, -Y_\ell) \ll 1$ for all hidden ℓ, meaning that Theorem 3 ensures the preservation of each J_ℓ spectrum for $\ell > \mathscr{L}_{\text{neg}}$ through negation. For $\ell \leq \mathscr{L}_{\text{neg}}$, gradients are non-exploding but possibly vanishing. Hence, **C2a**. Theorem 4 provides **C2b** in many cases but does not cover ReLU-like activations, we conjecture that negation is "approximately" LWOP in this case.

5.3. Selecting Layers for Negation

Generally, we only negate the weights of the first layer, as this is sufficient to induce changes in low-level feature representations, leading to significant parameter updates in deeper layers during fine-tuning. As a result, high-level features containing user-specific information are effectively forgotten. Additionally, while negating multiple layers can strengthen unlearning, it also slows down recovery. For instance, consider negating two layers ℓ_1 and ℓ_2, where $\ell_1 < \ell_2$ in the computational graph. On one hand, Theorem 4 suggests that the simplest recovery path involves modifying layers $\ell > \ell_1$. On the other hand, Theorem 3 indicates that layers $\ell \leq \ell_2$ may suffer from vanishing gradients. Consequently, layers in the range $\{\ell_1 < \ell \leq \ell_2\} \neq \emptyset$ should be fine-tuned but are likely to have small gradients, making recovery slower. Additionally, negating both convolution and normalization layers sequentially is ineffective, as the negations cancel out[4].

6. Experiments

In this section, we evaluate NoT across three datasets and three model architectures within federated settings, benchmarking it against seven baseline methods. The main experiments cover random data forgetting, backdoor attack mitigation, and empirical validation of our theoretical predictions. Additionally, we evaluate NoT within a centralized setting, benchmarking it against eight baselines. Results indicate that NoT achieves superior unlearning performance with low communication and computation costs. Additionally, we present an ablation study examining the impact of negating different layers of a model, various perturbations, and changing the ratio of data to forget.

6.1. Experimental Setup

Datasets and Models. We evaluate NoT using CIFAR-10/100 [35] and Caltech-101 [36], with three architectures: CNN (two convolution layers with layer normalization), ResNet-18 [22], and Vision Transformer (ViT-B/16) [9].

Implementation Details. For each dataset and architecture, we train a global model using FedAvg until convergence, then apply the unlearning algorithm to the converged model. Each communication round involves the participation of all clients, with data distributed IID among them unless stated otherwise. Each client's data is divided into training and validation sets (80:20), while the test set is used to assess model accuracy. For Caltech-101, we initially split 80% of samples for training and 20% for testing, then partition the training data among clients. Further details are provided in Appendix 14.

Baselines. We compare NoT with several baselines: ❶ *Retrain*: Retraining from scratch with the retain data D_r, serving as the gold standard model due to exact unlearning; ❷ *FT*: Fine-tuning the original converged model solely with D_r, relying on natural forgetting of D_u; ❸ *FedEraser* [38]; ❹ *FUKD* [63]; ❺ *PGD* [21]; ❻ *MoDE* [71]; and ❼ *FCU* [8].

Evaluation Metrics. Following prior works [13, 28], we assess NoT's effectiveness and efficiency with the following metrics: ❶ *Retain*, ❷ *Forget*, and ❸ *Test Accuracies (%)*, measuring the model's performance on D_r, D_u, and test

[3] By feedforward, we mean that the computational graph is a DAG. RNNs are not feedforward, but residual connections are allowed.

[4] Negating both the scale σ and mean μ in a normalization layer N along with the convolution C (without non-linearity) results in: $(-N)(-C)(x) = ((-Cx) - (-\mu))/(-\sigma) = (Cx - \mu)/\sigma = NC(x)$.

Table 1. **Client-wise federated unlearning in an IID setting.** Performance comparison of NoT with baselines in a 10-client setup, where the first client requests unlearning. The best average gap is marked in red.

Dataset & Model	Method	Accuracy (%)			Privacy (%)	Avg. Gap ↓	Cost (Bytes & FLOPs)	
		Retain (Δ ↓)	Forget (Δ ↓)	Test (Δ ↓)	MIA (Δ ↓)		Comm. ↓	Comp. ↓
CIFAR-10 CNN	Retrain	$91.66_{\pm 0.12}$ (0.00)	$83.05_{\pm 0.23}$ (0.00)	$82.32_{\pm 0.30}$ (0.00)	$50.23_{\pm 0.39}$ (0.00)	0.00	$1.35e^{10}$	$5.81e^{16}$
	FT	$92.48_{\pm 0.20}$ (0.82)	$85.56_{\pm 0.36}$ (2.51)	$82.36_{\pm 0.08}$ (0.04)	$50.90_{\pm 0.71}$ (0.67)	1.01	$9.39e^{09}$	$4.06e^{16}$
	FedEraser	$88.19_{\pm 0.16}$ (3.47)	$81.71_{\pm 0.23}$ (1.34)	$80.87_{\pm 0.37}$ (1.45)	$50.17_{\pm 0.26}$ (0.06)	1.58	$1.34e^{10}$	$5.79e^{16}$
	FUKD	$82.69_{\pm 0.05}$ (8.97)	$79.31_{\pm 0.12}$ (3.74)	$78.71_{\pm 0.12}$ (3.61)	$50.17_{\pm 0.49}$ (0.06)	4.09	$1.33e^{10}$	$5.77e^{16}$
	PGD	$92.62_{\pm 0.13}$ (0.96)	$85.36_{\pm 0.30}$ (2.31)	$82.50_{\pm 0.02}$ (0.18)	$50.70_{\pm 0.45}$ (0.47)	0.98	$1.19e^{10}$	$5.13e^{16}$
	MoDE	$92.56_{\pm 0.13}$ (0.90)	$85.25_{\pm 0.62}$ (2.20)	$82.31_{\pm 0.35}$ (0.01)	$50.70_{\pm 0.41}$ (0.47)	0.90	$1.10e^{10}$	$4.77e^{16}$
	FCU	$92.46_{\pm 0.11}$ (0.80)	$84.84_{\pm 0.22}$ (1.79)	$82.48_{\pm 0.21}$ (0.16)	$50.70_{\pm 0.36}$ (0.47)	0.81	$1.33e^{10}$	$5.75e^{16}$
	NoT (Ours)	$91.69_{\pm 0.02}$ (0.03)	$83.86_{\pm 0.17}$ (0.81)	$82.65_{\pm 0.14}$ (0.33)	$50.23_{\pm 0.21}$ (0.00)	0.29	$7.09e^{09}$	$3.06e^{16}$
CIFAR-100 CNN	Retrain	$72.32_{\pm 0.11}$ (0.00)	$53.31_{\pm 0.87}$ (0.00)	$54.28_{\pm 0.25}$ (0.00)	$49.70_{\pm 0.64}$ (0.00)	0.00	$1.38e^{10}$	$5.96e^{16}$
	FT	$73.68_{\pm 0.06}$ (1.36)	$56.11_{\pm 0.45}$ (2.80)	$55.46_{\pm 0.08}$ (1.18)	$49.77_{\pm 1.11}$ (0.07)	1.35	$1.33e^{10}$	$5.77e^{16}$
	FedEraser	$67.25_{\pm 0.44}$ (5.07)	$51.02_{\pm 0.05}$ (2.29)	$51.51_{\pm 0.62}$ (2.77)	$49.60_{\pm 0.57}$ (0.10)	2.56	$1.38e^{10}$	$5.96e^{16}$
	FUKD	$55.99_{\pm 0.03}$ (16.33)	$45.20_{\pm 0.04}$ (8.11)	$47.32_{\pm 0.12}$ (6.96)	$51.13_{\pm 0.24}$ (1.43)	8.21	$1.38e^{10}$	$5.95e^{16}$
	PGD	$73.68_{\pm 0.11}$ (1.36)	$56.00_{\pm 0.47}$ (2.69)	$55.21_{\pm 0.07}$ (0.93)	$49.83_{\pm 0.95}$ (0.13)	1.28	$1.21e^{10}$	$5.22e^{16}$
	MoDE	$73.36_{\pm 0.45}$ (1.04)	$55.64_{\pm 0.37}$ (2.33)	$55.16_{\pm 0.19}$ (0.88)	$49.67_{\pm 0.94}$ (0.03)	1.07	$1.20e^{10}$	$5.19e^{16}$
	FCU	$73.40_{\pm 0.11}$ (1.08)	$56.68_{\pm 0.08}$ (3.37)	$55.37_{\pm 0.08}$ (1.09)	$50.03_{\pm 0.19}$ (0.33)	1.47	$1.04e^{10}$	$4.49e^{16}$
	NoT (Ours)	$72.25_{\pm 0.08}$ (0.07)	$55.22_{\pm 0.61}$ (1.91)	$55.23_{\pm 0.39}$ (0.95)	$49.63_{\pm 0.97}$ (0.07)	0.75	$1.33e^{10}$	$5.73e^{16}$
CIFAR-10 ResNet-18	Retrain	$100.00_{\pm 0.00}$ (0.00)	$87.66_{\pm 0.64}$ (0.00)	$87.73_{\pm 0.35}$ (0.00)	$49.37_{\pm 0.29}$ (0.00)	0.00	$1.23e^{12}$	$5.66e^{18}$
	FT	$99.45_{\pm 0.77}$ (0.55)	$97.36_{\pm 2.22}$ (9.70)	$86.61_{\pm 1.54}$ (1.12)	$56.87_{\pm 0.84}$ (7.50)	4.72	$5.94e^{11}$	$2.73e^{18}$
	PGD	$99.51_{\pm 0.69}$ (0.49)	$97.67_{\pm 2.48}$ (10.01)	$86.82_{\pm 1.89}$ (0.91)	$56.70_{\pm 1.31}$ (7.33)	4.68	$6.04e^{11}$	$2.78e^{18}$
	MoDE	$99.80_{\pm 0.20}$ (0.20)	$91.18_{\pm 0.68}$ (3.52)	$87.11_{\pm 0.65}$ (0.62)	$52.37_{\pm 0.66}$ (3.00)	1.83	$5.88e^{11}$	$2.71e^{18}$
	FCU	$100.00_{\pm 0.00}$ (0.00)	$87.51_{\pm 0.45}$ (0.15)	$85.93_{\pm 0.07}$ (1.80)	$50.80_{\pm 0.28}$ (1.43)	0.84	$5.73e^{11}$	$2.64e^{18}$
	NoT (Ours)	$99.77_{\pm 0.31}$ (0.23)	$91.62_{\pm 2.06}$ (3.96)	$87.63_{\pm 1.61}$ (0.10)	$52.20_{\pm 0.83}$ (2.83)	1.78	$5.42e^{11}$	$2.49e^{18}$
CIFAR-100 ResNet-18	Retrain	$99.96_{\pm 0.00}$ (0.00)	$59.96_{\pm 0.61}$ (0.00)	$60.66_{\pm 0.63}$ (0.00)	$50.30_{\pm 0.30}$ (0.00)	0.00	$7.34e^{11}$	$3.38e^{18}$
	FT	$99.85_{\pm 0.12}$ (0.11)	$88.80_{\pm 3.22}$ (28.84)	$60.41_{\pm 1.77}$ (0.25)	$64.33_{\pm 1.10}$ (14.03)	10.81	$7.28e^{11}$	$3.35e^{18}$
	PGD	$99.49_{\pm 0.41}$ (0.47)	$75.30_{\pm 1.35}$ (15.34)	$61.12_{\pm 0.42}$ (0.46)	$57.33_{\pm 0.73}$ (7.03)	5.83	$6.67e^{11}$	$3.07e^{18}$
	MoDE	$99.69_{\pm 0.29}$ (0.27)	$73.53_{\pm 3.62}$ (13.57)	$60.74_{\pm 2.33}$ (0.08)	$57.00_{\pm 2.02}$ (6.70)	5.16	$6.76e^{11}$	$3.12e^{18}$
	FCU	$99.92_{\pm 0.03}$ (0.04)	$67.14_{\pm 0.20}$ (7.18)	$60.52_{\pm 0.26}$ (0.14)	$52.97_{\pm 0.76}$ (2.67)	2.51	$3.70e^{11}$	$1.71e^{18}$
	NoT (Ours)	$99.35_{\pm 0.64}$ (0.61)	$72.03_{\pm 4.20}$ (12.07)	$61.98_{\pm 2.47}$ (1.32)	$55.20_{\pm 0.80}$ (4.90)	4.73	$6.09e^{11}$	$2.80e^{18}$
Caltech-101 ViT	Retrain	$99.73_{\pm 0.04}$ (0.00)	$48.29_{\pm 0.44}$ (0.00)	$48.02_{\pm 0.72}$ (0.00)	$49.67_{\pm 3.47}$ (0.00)	0.00	$1.76e^{12}$	$1.37e^{21}$
	FT	$99.96_{\pm 0.00}$ (0.23)	$94.23_{\pm 0.51}$ (45.94)	$48.75_{\pm 0.27}$ (0.73)	$73.80_{\pm 0.94}$ (24.13)	17.76	$1.63e^{12}$	$1.28e^{21}$
	PGD	$73.34_{\pm 14.31}$ (26.39)	$61.44_{\pm 12.13}$ (13.15)	$44.22_{\pm 2.41}$ (3.80)	$61.43_{\pm 6.47}$ (11.76)	13.78	$4.78e^{10}$	$3.94e^{19}$
	MoDE	$99.82_{\pm 0.09}$ (0.09)	$52.01_{\pm 4.13}$ (3.72)	$48.02_{\pm 0.59}$ (0.00)	$52.27_{\pm 2.90}$ (2.60)	1.60	$1.73e^{12}$	$1.36e^{21}$
	FCU	$99.07_{\pm 0.10}$ (0.66)	$51.83_{\pm 0.61}$ (3.54)	$48.62_{\pm 0.33}$ (0.60)	$51.30_{\pm 0.22}$ (1.63)	1.61	$8.56e^{11}$	$8.36e^{20}$
	NoT (Ours)	$99.70_{\pm 0.02}$ (0.03)	$50.81_{\pm 0.73}$ (2.52)	$47.83_{\pm 0.27}$ (0.19)	$50.07_{\pm 2.04}$ (0.40)	0.79	$8.08e^{11}$	$6.31e^{20}$

data, respectively. ❹ *MIA* (Membership Inference Attack [26]) (%): indicates the extent to which D_u remains recognizable in the model ❺ *Communication* and ❻ *Computation Costs*: quantify the total communication (in bytes) and FLOPs needed for unlearning and recovery. Retain and test accuracies measure how well the model retains knowledge about D_r, while forget accuracy and MIA evaluate how effectively the model forgets D_u. We calculate delta (Δ) values and the average gap compared to Retrain, with lower values indicating better performance. Achieving a balance across all metrics is essential to demonstrate NoT's effectiveness and efficiency.

6.2. Results

Federated Unlearning: Comparing NoT with Baselines. Table 1 shows that NoT performance outperforms other baselines in client-wise FU across architectures and datasets, achieving a low average gap and competitive communication and computation costs. While FCU shows lower average gap in some cases, such as with ResNet-18, its performance varies across architectures, unlike NoT's consistent balance between forgetting effectiveness (Forget Accuracy, MIA), model fidelity (Retain, Test Accuracy), and competitively low costs. Although Table 1 does not report storage, FedEraser and FUKD require considerable storage that grows with communication rounds, making them impractical for ResNet-18 and ViT due to storage constraints. In contrast, NoT requires no additional storage. FedEraser and FUKD also perform worse than Retrain because, in our setting, the target client joins the federation from the first round, and as those baselines need to refer back to the first checkpoint, this makes full retraining more efficient. Table 4 in Appendix 13.2 presents results for client-wise FU in a non-IID setting. Furthermore, results

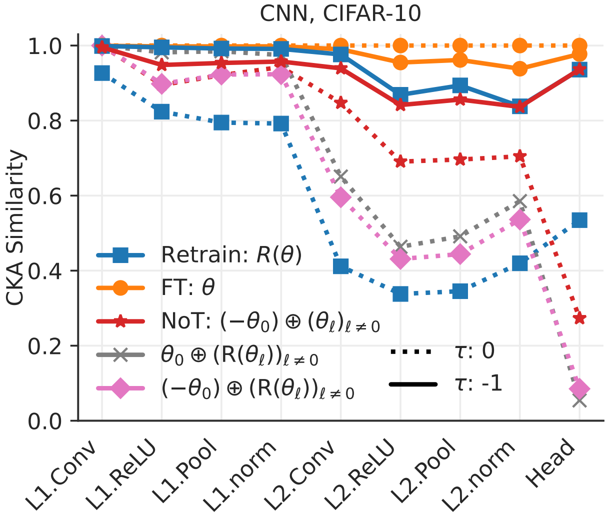

Figure 3. **CKA of layer activations** for various models compared to the original model ($\theta = \theta^*$) before fine-tuning (FT@τ:0). The first and last communication rounds are denoted by τ:0 and -1. $(-\theta_0) \oplus (\mathrm{R}(\theta_\ell))_{\ell \neq 0}$ denotes a model with negated first-layer weights (ℓ:0) and randomized rest $R(\cdot)$.

for class-wise and instance-wise forgetting are provided in Tables 5 and 6 in Appendices 13.3 and 13.4, respectively. These results highlight NoT's superior performance across different settings, demonstrating its effectiveness and efficiency compared to baseline methods. Lastly, Table 7 in Appendix 13.5 shows that NoT achieves the lowest average gap with minimal communication and computation costs, outperforming baselines in removing backdoor influence.

Negation is LWOP and Breaks Co-Adaptation. We assess the impact of negation through LWOP and the disruption of co-adaptation by comparing CKA similarities [34] across three models—FT, Retrain, and NoT—analyzed before (τ:0) and after (τ:-1) fine-tuning (see Figure 3). The FT model at τ:0 serves as the reference. Initially comparing NoT to FT at τ:0; we observe a high CKA similarity at the negated layer (L1.conv), which is consistent with LWO of NoT model at τ:0, hence LWOP of negation. However, as layer depth increases, divergence grows, indicating a breakdown in co-adaptation. Post fine-tuning, both Retrain and NoT at τ:-1 exhibit similar CKA, diverging from FT but closely aligning with each other, underscoring the resemblance of NoT to Retrain. CKA comparisons between models obtained by randomizing (using a consistent seed) all layers except the first in both FT and NoT at τ:0 are also performed: the observed high CKA similarity further supports LWO. See Appendix 13.6 for extra CKA comparisons. Direct validation of LWO (Appendix 13.7) involves freezing and negating the first layer, while randomizing the others, then fine-tuning; this yields a model with test accuracy comparable to training from scratch, consistent with our predictions. Finally, the dimensionality reduction of the gradient descent search space is confirmed using PCA (Appendix 13.8).

Centralized Unlearning: Comparing NoT with Baselines. To validate NoT in a centralized setting, we compare it to baselines: ❶ *Retrain*; ❷ *FT*; ❸ *RandL* (Random Label): The forget set (D_u) is randomly relabeled, followed by fine-tuning the model on the updated dataset; ❹ *GA* (Gradient Ascent) [59]: Fine-tuning on D_u by increasing loss; ❺ *BadT* [6]; ❻ *ℓ_1-sparse* [28]; ❼ *SSD* [14]; and ❽ *SalUn* [13]. Table 2 confirms that NoT achieves the best balance of efficiency and accuracy without extra storage or access to D_u. While ℓ_1-sparse[5] achieves lower average gap, it requires more than 30% more computations. These results affirm NoT's broader applicability beyond federated settings.

6.3. Ablation Study

Negation of Different Layers. Figure 4 shows that negating ViT's convolution projection layer achieves the best performance. Negating the positional embedding layer initially raises average gap but eventually recovers, resembling natural forgetting (FT). Negating the query weights of the first self-attention head is effective but falls short of the projection layer. These findings suggest that early-layer negation is most effective, as the largest forgetting occurs in subsequent layers in the computational graph[6] due to a *break in co-adaptation*. This is evident in Figure 4, where negating the second self-attention head's query weights results in significantly less unlearning than the first.

Different Perturbations. Figure 5 evaluates various perturbations on ViT's convolution projection layer. Weight negation achieves the lowest average gap, while other perturbations (*e.g.*, random initialization, adding noise) perform similarly to FT (no perturbation). Kernel flips or rotations are likely LWOP and provide slight improvement but remain less effective than negation. Future research could investigate alternative LWOP perturbations with the potential for further enhancing unlearning performance.

Different Forget Data (D_u) Ratios. Figure 6 examines NoT's performance across varying forget ratios (*i.e.*, more target clients). Forget ratios of 10%, 50%, and 90% represent 1, 5, and 9 clients (out of 10) requesting unlearning, respectively. For all ratios, a Retrain model is computed using the new corresponding retain data. Across all ratios,

[5] Our theory applies to ℓ_1-sparse [28] as its sparsification step is a perturbation satisfying C1 and C2.

[6] With residual connections, the last layer may 'follow' the first layer.

Table 2. **Centralized unlearning.** Performance comparison of NoT with baselines, with the best average gap marked in red. The forget data is randomly selected and constitutes 10% of the train data.

Dataset & Model	**Method**	**Accuracy (%)**			**Privacy (%)**	**Avg. Gap** ↓	**Comp. Cost (FLOPs)** ↓
		Retain (Δ ↓)	**Forget (Δ ↓)**	**Test (Δ ↓)**	**MIA (Δ ↓)**		
CIFAR-10 ResNet-18	Retrain	$100.00_{\pm 0.00}$ (0.00)	$91.72_{\pm 0.14}$ (0.00)	$92.03_{\pm 0.12}$ (0.00)	$49.32_{\pm 0.23}$ (0.00)	0.00	$9.58e^{15}$
	FT	$98.63_{\pm 0.24}$ (1.37)	$95.92_{\pm 0.53}$ (4.20)	$89.84_{\pm 0.27}$ (2.19)	$53.49_{\pm 0.45}$ (4.17)	2.98	$7.31e^{14}$
	RandL	$94.34_{\pm 0.32}$ (5.66)	$88.85_{\pm 0.46}$ (2.87)	$89.53_{\pm 0.21}$ (2.50)	$54.60_{\pm 0.43}$ (5.28)	4.07	$2.07e^{15}$
	GA (EuroS&P, 2022)	$98.76_{\pm 0.07}$ (1.24)	$93.39_{\pm 0.19}$ (1.67)	$89.73_{\pm 0.14}$ (2.30)	$52.21_{\pm 0.24}$ (2.89)	2.02	$2.09e^{15}$
	BadT (AAAI, 2023)	$99.89_{\pm 0.03}$ (0.11)	$98.52_{\pm 0.30}$ (6.80)	$90.14_{\pm 0.17}$ (1.89)	$48.82_{\pm 0.12}$ (0.50)	2.32	$5.69e^{14}$
	ℓ_1-sparse (NIPS, 2023)	$99.98_{\pm 0.00}$ (0.02)	$91.94_{\pm 0.03}$ (0.22)	$92.15_{\pm 0.21}$ (0.12)	$49.63_{\pm 0.07}$ (0.31)	0.19	$2.98e^{15}$
	SSD (AAAI, 2024)	$98.82_{\pm 0.85}$ (1.18)	$99.05_{\pm 0.72}$ (7.33)	$89.70_{\pm 1.40}$ (2.33)	$58.25_{\pm 1.51}$ (8.93)	4.94	$7.46e^{13}$
	SalUn (ICLR, 2024)	$98.29_{\pm 0.19}$ (1.71)	$94.01_{\pm 0.45}$ (2.29)	$90.78_{\pm 0.13}$ (1.25)	$48.18_{\pm 1.34}$ (1.14)	1.64	$1.22e^{15}$
	NoT (Ours)	$99.69_{\pm 0.05}$ (0.31)	$92.41_{\pm 0.12}$ (0.69)	$92.18_{\pm 0.07}$ (0.15)	$49.52_{\pm 0.11}$ (0.20)	0.34	$2.19e^{15}$

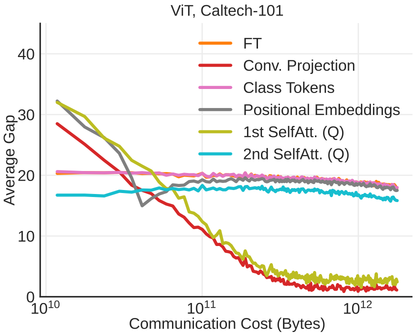

Figure 4. **Effect of negating different ViT layers.** Negating the convolution projection layer resulted in the best unlearning performance.

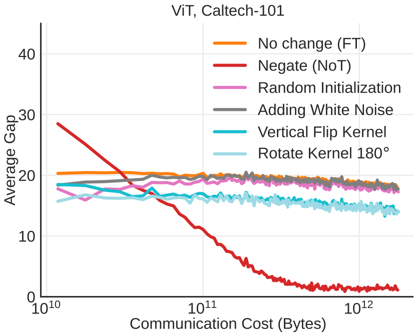

Figure 5. **Effect of different perturbations on ViT convolution projection layer.** Applying weight negation is the best perturbation for inducing unlearning.

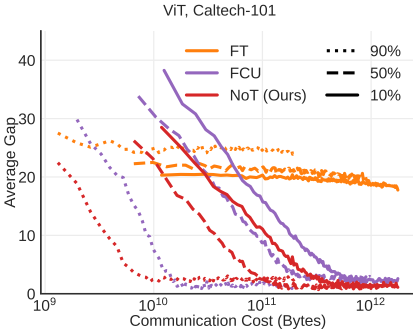

Figure 6. **Effect of varying forget data ratios (*i.e.*, target clients).** NoT attains the best unlearning with the least cost compared to baselines under different ratios.

NoT closely matches Retrain's performance with minimal communication costs. This is due to the reduction in retain data as forget data increases. In contrast, FCU's computation cost rises significantly as forget ratios increase, while NoT remains efficient due to not needing access to D_u.

7. Limitations

While NoT effectively unlearns targeted data contributions, it also leads to some loss of useful knowledge, necessitating access to retain data for recovery—similar to other strong baseline methods. This access to retain data is essential for optimal model performance. Moreover, NoT has been evaluated only on vision-centric classification tasks with smaller datasets (CIFAR-10/100, Caltech-101), and has not been tested on larger-scale datasets, such as Imagenet [52].

8. Conclusion

We proposed NoT, a novel and efficient federated unlearning (FU) algorithm that requires no additional storage or access to target data. NoT achieves unlearning by negating specific layer-wise parameters, disrupting co-adaptation across layers. We also proposed a theoretical framework, supported by empirical results, that demonstrates that layer-wise negation effectively induces unlearning, while the model's good optimization state post-negation enables rapid recovery during fine-tuning. NoT has shown strong unlearning and recovery performance across diverse architectures and datasets, surpassing existing FU baselines and outperforming traditional machine unlearning methods.

References

[1] SI Amari. Information geometry. *Contemporary Mathematics*, 203:81–96, 1997. 4

[2] Theodore Wilbur Anderson. Asymptotic theory for principal component analysis. *The Annals of Mathematical Statistics*, 34(1):122–148, 1963. 9

[3] Thomas Bachlechner, Bodhisattwa Prasad Majumder, Henry Mao, Gary Cottrell, and Julian McAuley. Rezero is all you need: Fast convergence at large depth. In *Uncertainty in Artificial Intelligence*, pages 1352–1361. PMLR, 2021. 4

[4] Rebekka Burkholz and Alina Dubatovka. Initialization of relus for dynamical isometry. *Advances in Neural Information Processing Systems*, 32, 2019. 4

[5] Minmin Chen, Jeffrey Pennington, and Samuel Schoenholz. Dynamical isometry and a mean field theory of rnns: Gating enables signal propagation in recurrent neural networks. In

International Conference on Machine Learning, pages 873–882. PMLR, 2018. 4

[6] Vikram S Chundawat, Ayush K Tarun, Murari Mandal, and Mohan Kankanhalli. Can bad teaching induce forgetting? unlearning in deep networks using an incompetent teacher. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(6):7210–7217, 2023. 7

[7] Donald L Cohn. *Measure theory*. Springer, 2013. 5

[8] Zhipeng Deng, Luyang Luo, and Hao Chen. Enable the right to be forgotten with federated client unlearning in medical imaging. *arXiv preprint arXiv:2407.02356*, 2024. 2, 5

[9] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 5

[10] Dumitru Erhan, Pierre-Antoine Manzagol, Yoshua Bengio, Samy Bengio, and Pascal Vincent. The difficulty of training deep architectures and the effect of unsupervised pre-training. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, pages 153–160, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA, 2009. PMLR. 4

[11] Dumitru Erhan, Aaron Courville, Yoshua Bengio, and Pascal Vincent. Why does unsupervised pre-training help deep learning? In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 201–208. JMLR Workshop and Conference Proceedings, 2010. 4

[12] European Union EU. Complete guide to general data protection regulation compliance. *https://gdpr.eu/*, 2023. 1

[13] Chongyu Fan, Jiancheng Liu, Yihua Zhang, Eric Wong, Dennis Wei, and Sijia Liu. Salun: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. In *The Twelfth International Conference on Learning Representations*, 2024. 5, 7

[14] Jack Foster, Stefan Schoepf, and Alexandra Brintrup. Fast machine unlearning without retraining through selective synaptic dampening. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(11):12043–12051, 2024. 3, 7

[15] D.J. Futuyma. Coevolution. In *Brenner's Encyclopedia of Genetics (Second Edition)*, pages 70–75. Academic Press, San Diego, second edition edition, 2013. 1

[16] Xiangshan Gao, Xingjun Ma, Jingyi Wang, Youcheng Sun, Bo Li, Shouling Ji, Peng Cheng, and Jiming Chen. Verifi: Towards verifiable federated unlearning. *IEEE Transactions on Dependable and Secure Computing*, 2024. 2

[17] Stuart Geman. A limit theorem for the norm of random matrices. *The Annals of Probability*, 8(2):252–261, 1980. 9

[18] Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9304–9312, 2020. 2

[19] Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Forgetting outside the box: Scrubbing deep networks of information accessible from input-output observations. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX 16*, pages 383–398. Springer, 2020. 3

[20] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain, 2019. 8

[21] Anisa Halimi, Swanand Kadhe, Ambrish Rawat, and Nathalie Baracaldo. Federated unlearning: How to efficiently erase a client in fl? *arXiv preprint arXiv:2207.05521*, 2022. 1, 2, 5, 8

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[23] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4918–4927, 2019. 4

[24] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012. 2, 1

[25] Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116, 1998. 4

[26] Hongsheng Hu, Zoran Salcic, Lichao Sun, Gillian Dobbie, Philip S Yu, and Xuyun Zhang. Membership inference attacks on machine learning: A survey. *ACM Computing Surveys (CSUR)*, 54(11s):1–37, 2022. 6

[27] Arthur Ulysse Jacot-Guillarmod. Theory of deep learning: Neural tangent kernel and beyond. Technical report, EPFL, 2022. 4

[28] Jinghan Jia, Jiancheng Liu, Parikshit Ram, Yuguang Yao, Gaowen Liu, Yang Liu, Pranay Sharma, and Sijia Liu. Model sparsity can simplify machine unlearning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 5, 7

[29] Ryo Karakida, Shotaro Akaho, and Shun-ichi Amari. Universal statistics of fisher information in deep neural networks: Mean field approach. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, pages 1032–1041. PMLR, 2019. 4

[30] Noureddine El Karoui. Tracy–Widom limit for the largest eigenvalue of a large class of complex sample covariance matrices. *The Annals of Probability*, 35(2):663 – 714, 2007. 9

[31] Noureddine El Karoui. Spectrum estimation for large dimensional covariance matrices using random matrix theory. *The Annals of Statistics*, 36(6):2757 – 2790, 2008. 10

[32] Yasser H. Khalil, Amir Hossein Estiri, Mahdi Beitollahi, Nader Asadi, Sobhan Hemati, Xu Li, Guojun Zhang, and Xi Chen. DFML: Decentralized federated mutual learning. *Transactions on Machine Learning Research*, 2024. 1

[33] Daijin Kim and Sunha Ahn. An optimal vq codebook design using the co-adaptation of learning and evolution. In *Soft Computing in Industrial Applications*, pages 225–239, London, 2000. Springer London. 1

[34] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *Proceedings of the 36th International*

Conference on Machine Learning, pages 3519–3529. PMLR, 2019. 7

[35] Alex Krizhevsky, Vinod Nair, Geoffrey Hinton, et al. The cifar-10 dataset. *online: http://www. cs. toronto. edu/kriz/cifar. html*, 55(5):2, 2014. 5

[36] Fei-Fei Li, Marco Andreeto, Marc'Aurelio Ranzato, and Pietro Perona. Caltech 101, 2022. 5

[37] Zhibin Liao, Tom Drummond, Ian Reid, and Gustavo Carneiro. Approximate fisher information matrix to characterize the training of deep neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 42(1): 15–26, 2018. 4

[38] Gaoyang Liu, Xiaoqiang Ma, Yang Yang, Chen Wang, and Jiangchuan Liu. Federaser: Enabling efficient client-level data removal from federated learning models. In *2021 IEEE/ACM 29th international symposium on quality of service (IWQOS)*, pages 1–10. IEEE, 2021. 1, 2, 5

[39] Ziyao Liu, Yu Jiang, Jiyuan Shen, Minyi Peng, Kwok-Yan Lam, and Xingliang Yuan. A survey on federated unlearning: Challenges, methods, and future directions. *arXiv preprint arXiv:2310.20448*, 2023. 1

[40] Ziyao Liu, Huanyi Ye, Chen Chen, Yongsen Zheng, and Kwok-Yan Lam. Threats, attacks, and defenses in machine unlearning: A survey. *IEEE Open Journal of the Computer Society*, 2025. 1

[41] Josh S Merel, Roy Fox, Tony Jebara, and Liam Paninski. A multi-agent control framework for co-adaptation in brain-computer interfaces. *Advances in Neural Information Processing Systems*, 26, 2013. 1

[42] James A Mingo and Roland Speicher. *Free probability and random matrices*. Springer, 2017. 10

[43] Dinh C Nguyen, Ming Ding, Pubudu N Pathirana, Aruna Seneviratne, Jun Li, and H Vincent Poor. Federated learning for internet of things: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 23(3):1622–1658, 2021. 1

[44] Maria-Irina Nicolae, Mathieu Sinn, Minh Ngoc Tran, Beat Buesser, Ambrish Rawat, Martin Wistuba, Valentina Zantedeschi, Nathalie Baracaldo, Bryant Chen, Heiko Ludwig, Ian M. Molloy, and Ben Edwards. Adversarial robustness toolbox v1.0.0, 2019. 8

[45] Lorenzo Noci, Sotiris Anagnostidis, Luca Biggio, Antonio Orvieto, Sidak Pal Singh, and Aurelien Lucchi. Signal propagation in transformers: Theoretical perspectives and the role of rank collapse. *Advances in Neural Information Processing Systems*, 35:27198–27211, 2022. 4

[46] Zibin Pan, Zhichao Wang, Chi Li, Kaiyan Zheng, Boqi Wang, Xiaoying Tang, and Junhua Zhao. Federated unlearning with gradient descent and conflict mitigation. *arXiv preprint arXiv:2412.20200*, 2024. 2

[47] Jeffrey Pennington, Samuel Schoenholz, and Surya Ganguli. Resurrecting the sigmoid in deep learning through dynamical isometry: theory and practice. *Advances in neural information processing systems*, 30, 2017. 4

[48] Magnus Rattray, David Saad, and Shun-ichi Amari. Natural gradient descent for on-line learning. *Physical review letters*, 81(24):5461, 1998. 4

[49] Chao Ren, Han Yu, Hongyi Peng, Xiaoli Tang, Anran Li, Yulan Gao, Alysa Ziying Tan, Bo Zhao, Xiaoxiao Li, Zengxiang Li, et al. Advances and open challenges in federated learning with foundation models. *arXiv preprint arXiv:2404.15381*, 2024. 2

[50] Daniel Revuz and Marc Yor. *Continuous martingales and Brownian motion*. Springer Science & Business Media, 2013. 2

[51] Nicolò Romandini, Alessio Mora, Carlo Mazzocca, Rebecca Montanari, and Paolo Bellavista. Federated unlearning: A survey on methods, design guidelines, and evaluation metrics. *IEEE Transactions on Neural Networks and Learning Systems*, 2024. 1

[52] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015. 8

[53] Ikuro Sato, Kohta Ishikawa, Guoqing Liu, and Masayuki Tanaka. Breaking inter-layer co-adaptation by classifier anonymization. *arXiv preprint arXiv:1906.01150*, 2019. 2, 1

[54] Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013. 4

[55] Zhiguo Shi, Jun Tu, Qiao Zhang, Lei Liu, and Junming Wei. A survey of swarm robotics system. In *Advances in Swarm Intelligence: Third International Conference, ICSI 2012, Shenzhen, China, June 17-20, 2012 Proceedings, Part I 3*, pages 564–572. Springer, 2012. 1

[56] Mandt Stephan, Matthew D Hoffman, David M Blei, et al. Stochastic gradient descent as approximate bayesian inference. *Journal of Machine Learning Research*, 18(134):1–35, 2017. 3, 4, 7

[57] Wojciech Tarnowski, Piotr Warchoł, Stanisław Jastrzobski, Jacek Tabor, and Maciej Nowak. Dynamical isometry is achieved in residual networks in a universal way for any activation function. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2221–2230. PMLR, 2019. 4

[58] Ayush K Tarun, Vikram S Chundawat, Murari Mandal, and Mohan Kankanhalli. Fast yet effective machine unlearning. *IEEE Transactions on Neural Networks and Learning Systems*, 2023. 3

[59] Anvith Thudi, Gabriel Deza, Varun Chandrasekaran, and Nicolas Papernot. Unrolling sgd: Understanding factors influencing machine unlearning. In *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*, pages 303–319. IEEE, 2022. 7

[60] Cédric Villani. *Topics in optimal transportation*. American Mathematical Soc., 2021. 4, 5

[61] Yichen Wan, Youyang Qu, Wei Ni, Yong Xiang, Longxiang Gao, and Ekram Hossain. Data and model poisoning backdoor attacks on wireless federated learning, and the defense mechanisms: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 2024. 1

[62] Mingjie Wang, Hao Cai, Xian-Feng Han, Jun Zhou, and Minglun Gong. Stnet: Scale tree network with multi-level auxiliator for crowd counting. *IEEE Transactions on Multimedia*, 25:2074–2084, 2022. 1

[63] Chen Wu, Sencun Zhu, and Prasenjit Mitra. Federated unlearning with knowledge distillation. *arXiv preprint arXiv:2201.09441*, 2022. 1, 2, 5, 8

[64] Lechao Xiao, Yasaman Bahri, Jascha Sohl-Dickstein, Samuel Schoenholz, and Jeffrey Pennington. Dynamical isometry and a mean field theory of CNNs: How to train 10,000-layer vanilla convolutional neural networks. In *Proceedings of the 35th International Conference on Machine Learning*, pages 5393–5402. PMLR, 2018. 4

[65] Jie Xu, Zihan Wu, Cong Wang, and Xiaohua Jia. Machine unlearning: Solutions and challenges. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2024. 1

[66] Yu Yao, Baosheng Yu, Chen Gong, and Tongliang Liu. Understanding how pretraining regularizes deep learning algorithms. *IEEE Transactions on Neural Networks and Learning Systems*, 34(9):5828–5840, 2021. 4

[67] Yong-Qua Yin, Zhi-Dong Bai, and Pathak R Krishnaiah. On the limit of the largest eigenvalue of the large dimensional sample covariance matrix. *Probability theory and related fields*, 78:509–521, 1988. 9

[68] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *Advances in neural information processing systems*, 27, 2014. 4

[69] Lefeng Zhang, Tianqing Zhu, Haibin Zhang, Ping Xiong, and Wanlei Zhou. Fedrecovery: Differentially private machine unlearning for federated learning frameworks. *IEEE Transactions on Information Forensics and Security*, 18: 4732–4746, 2023. 2

[70] Tuo Zhang, Lei Gao, Chaoyang He, Mi Zhang, Bhaskar Krishnamachari, and A Salman Avestimehr. Federated learning for the internet of things: Applications, challenges, and opportunities. *IEEE Internet of Things Magazine*, 5(1):24–29, 2022. 1

[71] Yian Zhao, Pengfei Wang, Heng Qi, Jianguo Huang, Zongzheng Wei, and Qiang Zhang. Federated unlearning with momentum degradation. *IEEE Internet of Things Journal*, 2023. 1, 2, 5, 8