Sequential Informed Federated Unlearning: Efficient and Provable Client Unlearning in Federated Optimization

Yann Fraboni, Richard Vidal, Laetitia Kameni, Marco Lorenzi

Sequential Informed Federated Unlearning: Efficient and Provable Client Unlearning in Federated Optimization

Yann Fraboni[1,2] **Richard Vidal**[2] **Laetitia Kameni**[2] **Marco Lorenzi**[1]
[1] Université Côte d'Azur, Inria Sophia Antipolis, Epione Research Group, France
and [2] Accenture Labs, Sophia Antipolis, France

Abstract

The aim of Machine Unlearning (MU) is to provide theoretical guarantees on the removal of the contribution of a given data point from a training procedure. Federated Unlearning (FU) consists in extending MU to unlearn a given client's contribution from a federated training routine. Current FU approaches are generally not scalable, and do not come with sound theoretical quantification of the effectiveness of unlearning. In this work we present Informed Federated Unlearning (IFU), a novel efficient and quantifiable FU approach. Upon unlearning request from a given client, IFU identifies the optimal FL iteration from which FL has to be reinitialized, with unlearning guarantees obtained through a randomized perturbation mechanism. The theory of IFU is also extended to account for sequential unlearning requests. Experimental results on different tasks and dataset show that IFU leads to more efficient unlearning procedures as compared to basic re-training and state-of-the-art FU approaches.

1 Introduction

With the emergence of new data regulations, such as the EU General Data Protection Regulation (GDPR) (Voigt and Von dem Bussche, 2017) and the California Consumer Privacy Act (CCPA) (Harding et al., 2019), the storage and processing of sensitive personal data is often subject of strict constraints and restrictions. In particular, the "right to be forgotten" states that personal data must be erased upon request from the concerned individuals, with subsequent potential implications on machine learning models trained by using this data. Machine Unlearning (MU) is an emerging discipline that studies methods to ideally remove the contribution of a given data instance used to train a machine learning model. Current MU approaches are essentially based on routines that modify the model weights in order to guarantee the "forgetting" of a given data point, i.e. to obtain a model equivalent to an hypothetical one trained without this data point (Cao and Yang, 2015; Bourtoule et al., 2021).

Motivated by data governance and confidentiality concerns, Federated learning (FL) has gained popularity in the last years to allow data owners to collaboratively learn a model without sharing their respective data. Among the different FL approaches, federated averaging (FEDAVG) has emerged as the most popular optimization scheme (McMahan et al., 2017). An optimization round of FEDAVG requires data owners, also called clients, to receive from the server the current global model which they update before sending it back to the server. The new global model is then created as the weighted average of the client updates, according to their data ratio. FL communication design guarantees to clients that their data is solely used to compute their model update, while theoretical work guarantees FL convergence to a stationary point of the clients' joint optimization problem (Wang et al., 2020; Li et al., 2020).

With the current deployments of FL in the real-world, it is of crucial importance to extend MU to guarantee the unlearning of clients wishing to opt-out from a collaborative training routine. This is not straightforward, since current MU schemes have been proposed essentially in the centralized learning setting, and cannot be seamlessly applied to the federated one. For example, several MU methods require the estimation of the Hessian of the loss function (Guo et al., 2020; Izzo et al., 2021; Golatkar et al., 2020a,b, 2021), an operation which is notoriously computationally heavy and intractable for high dimensional models. Moreover, sharing the Hessian would require clients to share with the server additional information about their data, thus exposing the federated setting to information leakage and attacks, for example under the form of model inversion (Fredrikson et al., 2015). Alternative MU methods draw from the concept of differential privacy Dwork and Roth (2014) and are based on a Gaussian noise perturbation of the trained model (Neel et al., 2021; Guo et al.,

2020; Gupta et al., 2021). The magnitude of the noise perturbation should be estimated directly from the clients data, which is by construction inaccessible to the server in the FL regime. We also note that while recent federated unlearning (FU) methods have been proposed to unlearn a client from the global FL model (Liu et al., 2021; Wang et al., 2021; Halimi et al., 2022; Wu et al., 2022), these approaches do not come with theoretical guarantees on the effectiveness of the unlearning.

The main contribution of this work consists in Informed Federated Unlearning (IFU), a novel efficient FU approach to unlearn a client's contribution with quantifiable unlearning guarantees. IFU requires minimal additional computations to the server in a standard FEDAVG procedure. Specifically, the server quantifies at every optimization round each client's contribution to the global model. Upon receiving an unlearning request from a client, the server identifies in the FL training history the optimal FL iteration and associated intermediate global model from which re-initializing the unlearning procedure. Unlearning guarantees are provided by introducing a novel randomized mechanism to perturb the selected intermediate model with client-specific noise. We also extend IFU to Sequential Informed Federated Unlearning (SIFU), to account for realistic unlearning scenarios where the server receives sequential unlearning requests from one or more clients at different times (Neel et al., 2021; Gupta et al., 2021).

This manuscript is structured as follows. In Section 2, we provide formal definitions for MU, FL, and FU, and introduce the randomized mechanism with associated unlearning guarantees. In Section 3, we introduce sufficient conditions for IFU to unlearn a client from the FL routine (Theorem 2). In Section 4, we extend IFU to the sequential unlearning setting with Sequential IFU (SIFU). Finally, in Section 5, we experimentally demonstrate on different tasks and datasets that SIFU leads to more efficient unlearning procedures as compared to basic re-training and state-of-the-art FU approaches.

2 Background and Related Work

In Section 2.1, we introduce the state-of-the art behind Machine Unlearning, while in Section 2.2, we introduce FL and FEDAVG. Finally, we introduce Federated Unlearning (FU) in Section 2.3.

2.1 Machine Unlearning

Let us consider a dataset $\mathcal{D}$ composed of two disjoint datasets: $\mathcal{D}_f$, the cohort of data samples on which unlearning must be applied after FL training, and $\mathcal{D}_k$, the remaining data samples. Hence, we have $\mathcal{D} = \mathcal{D}_f \cup \mathcal{D}_k$. We also consider $\mathcal{M}(\mathcal{D})$, the ML model parameters resulting from training with optimization scheme $\mathcal{M}$ on dataset $\mathcal{D}$. We introduce in this section the different unlearning baselines and methods currently used to unlearn $\mathcal{D}_f$ from the trained model $\mathcal{M}(\mathcal{D})$.

MU through retraining. Within this setting, a new training is performed from scratch with only $\mathcal{D}_k$ as training data. As the initial model contains no information from $\mathcal{D}_f$, the new trained model $\mathcal{M}(\mathcal{D}_k)$ also contains no information from $\mathcal{D}_f$. We note however that this procedure wastes the contribution of $\mathcal{D}_k$ already available by training originally on $\mathcal{D}$. Hence, this method is considered sub-optimal, and represents a basic baseline for unlearning approaches.

MU through fine-tuning. Fine-tuning on the remaining data $\mathcal{D}_k$ has been proposed as a practical approach to unlearn the specificities of $\mathcal{D}_f$. However, fine-tuning does not provide guarantees about the effectiveness of the unlearning. We provide an example of this issue in Appendix A.

MU through model scrubbing. Another unlearning approach consists in applying a "scrubbing" transformation h to the model $\mathcal{M}(\mathcal{D})$ such that the resulting model is as close as possible to the one that would be trained with only $\mathcal{D}_k$, i.e. $h(\mathcal{M}(\mathcal{D})) \approx \mathcal{M}(\mathcal{D}_k)$ (Ginart et al., 2019). To define a scrubbing method h, existing work mostly relies on the following Assumption 1, which considers a quadratic approximation of the loss function.

Assumption 1. *For model parameters $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, the gradient of the loss function of a given data point $\mathcal{D}_x$ satisfies*

$$\nabla f_{\mathcal{D}_x}(\boldsymbol{\phi}) = \nabla f_{\mathcal{D}_x}(\boldsymbol{\theta}) + H_{\mathcal{D}_x}(\boldsymbol{\theta})(\boldsymbol{\phi} - \boldsymbol{\theta}), \qquad (1)$$

where $H_{\mathcal{D}_x}(\boldsymbol{\theta})$ is positive semi-definite.

The scrubbed model is the new optimum obtained when unlearning data samples in $\mathcal{D}_f$. Hence, under Assumption 1, the new optimum can be obtained by setting $\nabla f_{\mathcal{D}_k}(h_{\mathcal{D}_k}(\boldsymbol{\theta})) = 0$, which gives

$$h_{\mathcal{D}_k}(\boldsymbol{\theta}) = \boldsymbol{\theta} - H^{-1}_{\mathcal{D}_k}(\boldsymbol{\theta}) \nabla f_{\mathcal{D}_k}(\boldsymbol{\theta}). \qquad (2)$$

With equation (2), h reduces to performing a Newton step, and has been derived in previous MU works (Guo et al., 2020; Izzo et al., 2021; Golatkar et al., 2020a,b, 2021; Mahadevan and Mathioudakis, 2021a) under different theoretical assumptions that can be generalized with Assumption 1. The main drawback behind the use of the scrubbing function (2) is the computation of the Hessian, which can be unfeasible for high dimensional model. Finally, the scrubbing function (2) is often coupled with Gaussian noise perturbation on the resulting weights (Golatkar et al., 2020a,b, 2021), to compensate the quadratic approximation of the loss function or the approximation of the Hessian.

MU through noise perturbation. This unlearning method consists in randomly perturbing the trained model $\mathcal{M}(\mathcal{D})$ to unlearn specificities from data samples in $\mathcal{D}_f$ (Neel et al., 2021; Gupta et al., 2021; Mahadevan and Mathioudakis,

2021b). The noise is set such that the guarantees of Definition 1 are satisfied, where (ϵ, δ) are parameters quantifying the unlearning guarantees.

Definition 1. *Let f_m be a randomized mechanism taking model parameters as an input. (ϵ, δ)-Unlearning trough f_m of a data point $\{x_m, y_m\}$ from a model $\mathcal{M}(D)$ is achieved if, for any subset $\mathcal{S}$ of the model parameters space and $D_{-m} \coloneqq D \setminus \{x_m, y_m\}$, we have*

$$\mathbb{P}(f_m(\mathcal{M}(D)) \in \mathcal{S}) \leq e^{\epsilon}\mathbb{P}(f_m(\mathcal{M}(D_{-m})) \in \mathcal{S}) + \delta \quad (3)$$

and $$\mathbb{P}(f_m(\mathcal{M}(D_{-m})) \in \mathcal{S}) \leq e^{\epsilon}\mathbb{P}(f_m(\mathcal{M}(D)) \in \mathcal{S}) + \delta. \quad (4)$$

Guo et al. (2020) shows the relationship between Definition 1 and the definition a randomized mechanism in differential privacy (Dwork and Roth, 2014; Chen et al., 2020).

2.2 Federated Optimization and FEDAVG

In FL, we consider a learning setup with M clients, and define $I = \{1, ..., M\}$ as the set of indices of the participating clients. Each client owns a dataset D_i composed of $|D_i| = n_i$ data samples. We consider a loss $f(\boldsymbol{x}_{i,l}, \boldsymbol{y}_{i,l}, \boldsymbol{\theta})$ assessed on each data sample $(\boldsymbol{x}_{i,l}, \boldsymbol{y}_{i,l}) \in D_i$, and define a client's loss function as $f_i(\boldsymbol{\theta}) \coloneqq 1/n_i \sum_{l=1}^{n_i} f(\boldsymbol{x}_{i,l}, \boldsymbol{y}_{i,l}, \boldsymbol{\theta})$. We define for the joint dataset $D_I \coloneqq \cup_{i \in I} D_i$ the federated loss function

$$f_I(\boldsymbol{\theta}) \coloneqq \frac{1}{|D_I|} \sum_{i \in I} |D_i| f_i(\boldsymbol{\theta}). \quad (5)$$

FEDAVG (McMahan et al., 2017) optimizes the loss (5) with theoretical guarantees for FL convergence to a stationary point (Wang et al., 2020; Li et al., 2020). At each iteration step n, the server sends the current global model parameters $\boldsymbol{\theta}^n$ to the clients. Each client updates the model by minimizing its local cost function $f_i(\boldsymbol{\theta})$ with K SGD steps initialized on $\boldsymbol{\theta}^n$. Subsequently each client returns the updated local parameters $\boldsymbol{\theta}_i^{n+1}$ to the server. The global model parameters $\boldsymbol{\theta}^{n+1}$ at the iteration step $n+1$ are then estimated as a weighted average, i.e.

$$\boldsymbol{\theta}^{n+1} = \boldsymbol{\theta}^n + \frac{1}{|D|} \sum_{i \in I} |D_i| \left[\boldsymbol{\theta}_i^{n+1} - \boldsymbol{\theta}^n\right]. \quad (6)$$

Algorithm 1 provides the implementation of FEDAVG. For the rest of this work, we define the joint dataset for a subset of client $I_x \subset I$ as $D_{I_x} \coloneqq \cup_{i \in I_x} D_i$.

2.3 Federated Unlearning

Existing works (Liu et al., 2021; Wang et al., 2021; Halimi et al., 2022; Wu et al., 2022) already consider the problem of unlearning a client from a model optimized through FEDAVG. However, these works do not provide theoretical nor quantitative guarantees on the unlearning procedure.

Algorithm 1 FEDAVG(I, N)

1: **for** n from 0 to $N-1$ **do**
2: The server sends $\boldsymbol{\theta}^n$ to every client in I.
3: Clients perform K SGDs to compute $\boldsymbol{\theta}_i^{n+1}$.
4: The server creates $\boldsymbol{\theta}^{n+1}$, equation (6).
5: **end for**
6: **return** the trained global model $\boldsymbol{\theta}^N$

Also, we note that standard MU methods cannot seamlessly be used in the federated setting. On one hand, federated unlearning (FU) with model scrubbing would require clients to perform only $K = 1$ SGD and share their Hessian with the server. Hence, model scrubbing decreases significantly FL convergence speed, while exposing the clients' data by sharing high order quantities with the risk of model inversion (Fredrikson et al., 2015). Moreover, the computation of the Hessian is unfeasible for highly dimensional models. On the other hand, existing MU approaches based on model perturbation require to retrain the model after the noise is added to the model's parameters. As such, retraining generally requires a significant amount of SGD steps to guarantee convergence to a new optimum, negatively affecting the effectiveness of the unlearning procedure.

In this work, we introduce a novel unlearning paradigm which avoids retraining the final model by identifying the optimal FL iteration where unlearning should be applied. Therefore, retraining is applied to an "early" version of the global model with reduced perturbation, thus minimizing the amount of required SGD steps to achieve convergence.

3 Unlearning a FL client with IFU

In this section, we develop our theory for the scenario where a model is trained with FEDAVG on the set of clients I, after which a client c requests unlearning of its own data. In Section 3.1, we define the sensitivity of the global model with respect to a client's contribution, and provide a bound relating this sensitivity to the FL procedure. In Section 3.2, we provide a tighter model sensitivity for some specific FL applications. Using Theorem 1, we introduce in Section 3.3 the perturbation procedure to unlearn a client c from the model trained with FEDAVG (Theorem 2). Finally, using Theorem 2, we introduce Informed Federated Unlearning (IFU) (Algorithm 2).

3.1 Theorem 1, Bounding the Model Sensitivity

As defined in Section 2.2, $\boldsymbol{\theta}_i^{n+1}$ is the local update of client i sent to the server after performing K SGD steps on its dataset D_i after initialization with global model $\boldsymbol{\theta}^n$. Given the contribution $\boldsymbol{\theta}_i^{n+1} - \boldsymbol{\theta}^n$ of a client i, we define the overall FL increment after aggregations across the set of clients

I as

$$\Delta(I, \boldsymbol{\theta}^n) \coloneqq \frac{1}{|D_I|} \sum_{i \in I} |D_i| \left[\boldsymbol{\theta}_i^{n+1} - \boldsymbol{\theta}^n \right]. \quad (7)$$

By comparing increments obtained by training on the set of clients I, and on the set $I_{-c} \coloneqq I \setminus \{c\}$ obtained after dropping a given client c, we define the bounded sensitivity after n server aggregations as

$$\Psi(n, c) \coloneqq \sum_{s=0}^{n-1} \left\| \Delta(I, \boldsymbol{\theta}^s) - \Delta(I_{-c}, \boldsymbol{\theta}^s) \right\|_2, \quad (8)$$

We show in Theorem 1 that the model sensitivity of FEDAVG can be bounded by the bounded sensitivity (8).

Theorem 1. *Under Assumption 1, the model sensitivity of* FEDAVG *when removing a client c after n server aggregations is defined as*

$$\alpha(n, c) \coloneqq \left\| \textsc{FedAvg}(I, n) - \textsc{FedAvg}(I_{-c}, n) \right\|_2, \quad (9)$$

where $\textsc{FedAvg}(I, n)$ *is the output of Algorithm 1, and*

$$\alpha(n, c) \leq \Psi(n, c). \quad (10)$$

Proof. See Appendix B. □

3.2 Improving the Tightness of the Sensitivity Bound

Theorem 1 shows that the bounded sensitivity provides a bound for the model sensitivity, while the computation of (8) only requires the clients' updated models, which are already shared with the server by design in FEDAVG. Nevertheless, we note that the bounded sensitivity (8) does not necessarily faithfully represent the evolution of the sensitivity across FL rounds. For instance, this quantity does not properly account for the unlearning of previous clients contributions for $s < n-1$. Indeed, these contributions should decrease across iterations due to the subsequent server aggregations and new clients' local work. To account for this aspect, we provide a tighter lower bound by assuming strongly convex and regularized local loss function, leading to a tighter bound for the model sensitivity of FEDAVG (Corollary 1).

Corollary 1. *Under Assumption 1, when considering that clients loss functions are μ-strongly convex and regularized with an L2 norm of weight λ, we have $\alpha(n, c) \leq \Psi(n, c)$ and*

$$\Psi(n, c) = \sum_{s=0}^{n-1} \left(1 - \eta(\lambda + \mu)\right)^{[(n-1)-s]K} \times \left\| \Delta(I, \boldsymbol{\theta}^s) - \Delta(I_{-c}, \boldsymbol{\theta}^s) \right\|_2, \quad (11)$$

where η and K are respectively the clients' local learning rate and amount of local work.

Proof. See Appendix B.3. □

The bounded sensitivity of Corollary 1 shows the following aspects. (1) The importance of a client's contribution decreases through aggregation rounds. (2) Since FL is guaranteed to converge to a stationary point (Wang et al., 2020; Li et al., 2020), so does the bounded sensitivity since $\lambda + \mu > 0$. (3) The bounded sensitivity is not necessarily inversely proportional to K. Indeed, due to data heterogeneity, with an increase in K every local model gets closer to its local optimum and the quantity $\left\| \Delta(I, \boldsymbol{\theta}^n) - \Delta(I_{-c}, \boldsymbol{\theta}^n) \right\|_2$ increases with the amount of local work K.

When clients have same data distribution, we retrieve $\Delta(I, \boldsymbol{\theta}^n) = \Delta(I_{-c}, \boldsymbol{\theta}^n)$, which yields null bounded sensitivity for every client, i.e. $\Psi(n, c) = 0$. We also note that the bound provided in Corollary 1 is tight, e.g. when considering identical eigenvalues for the Hessian of every local loss. More generally, the bound is tight in the limit case where the local learning rate of the clients is small.

We can draw an analogy between the bounded sensitivity (8) and client clustering in FL (Sattler et al., 2021; Fraboni et al., 2021a), where clients are clustered based on their contribution. In this work, the bounded sensitivity (8) is used instead to bound the sensitivity of the global model across rounds in FEDAVG.

3.3 Satisfying Definition 1

In this section, we introduce a randomized mechanism to provide guarantees for the unlearning of a given client c, where the magnitude of the perturbation process (Dwork and Roth, 2014) is defined based on the sensitivity of Theorem 1. In practice, we define a Gaussian noise mechanism to perturb each parameter of global model $\boldsymbol{\theta}^n$ such that we achieve (ϵ, δ)-unlearning of client c for the resulting model, according to Definition 1. We give in Theorem 2 sufficient conditions for the noise perturbation to satisfy Definition 1.

Theorem 2. *Defining*

$$\sigma(n, c) := \left[2 \left(\ln(1.25) - \ln(\delta)\right)\right]^{1/2} \epsilon^{-1} \Psi(n, c), \quad (12)$$

the noise perturbation $\sigma(n, c) \boldsymbol{I}_{\boldsymbol{\theta}}$ applied to the global model $\boldsymbol{\theta}^n$, where $\boldsymbol{I}_{\boldsymbol{\theta}}$ is the identity matrix, achieves (ϵ, δ)-unlearnig of client c according to Definition 1.

Proof. Follows directly from Theorem 1 coupled with Theorem A.1 of Dwork and Roth (2014). □

We note that, according to Theorem 2, (ϵ, δ)-unlearning a client from a given global model requires a standard deviation for the noise that is client-specific and proportional to its bounded sensitivity.

Algorithm 2 Informed Federated Unlearning (IFU)

DURING LEARNING WITH FEDAVG

FEDAVG(I, N) initialized on initial model $\boldsymbol{\theta}^0$.
for n from 0 to $N-1$, and i from 1 to c **do**
Compute $\Psi(n, i)$, equation (8).
end for

WHEN UNLEARNING CLIENT c

Require: c, ϵ, δ, σ, and amount of retraining steps $\tilde{N}$.
Get Ψ^* with equation (13).
Get $T = \arg\max_n (\Psi(n, c) \leq \Psi^*)$ with eq. (14).
The new global model is $\tilde{\boldsymbol{\theta}} = \boldsymbol{\theta}^T + N(0, \sigma^2 \boldsymbol{I}_{\boldsymbol{\theta}})$.
Run FEDAVG(I_{-c}, $\tilde{N}$) initialized on $\tilde{\boldsymbol{\theta}}$.

In what follows, the unlearning procedure will be defined with respect to the sensitivity threshold Ψ^* related to the unlearning budget (ϵ, δ) and standard deviation σ:

$$\Psi^* \coloneqq \left[2\left(\ln(1.25) - \ln(\delta)\right)\right]^{-1/2} \epsilon\sigma. \tag{13}$$

3.4 Informed Federated Unlearning (IFU)

Using the bounded sensitivity (8) and Theorem 2, we introduce Informed Federated Unlearning (IFU) to unlearn the contribution of client $c \in I$ from a FL training procedure based on FEDAVG. Algorithm 2 provides the implementation of IFU on top of FEDAVG. We note that during the FL training, IFU requires the server to compute the bounded sensitivity metric $\Psi(n, i)$ from each client's contribution $\boldsymbol{\theta}_i^{n+1}$ and current global model $\boldsymbol{\theta}^n$. These quantities are tracked throughout FL iterations and are used to identify the optimal unlearning strategy after request from a client c.

To unlearn client c, the server identifies the unlearning index T associated to the history of bounded sensitivity metrics, i.e. the most recent global model index such that a perturbation of size σ satisfies Theorem 2:

$$T \coloneqq \arg\max_n \left(\Psi(n, c) \leq \Psi^*\right). \tag{14}$$

The new global model is obtained after perturbation $\tilde{\boldsymbol{\theta}} \coloneqq \boldsymbol{\theta}^T + \nu$, where $\nu \sim N(0, \sigma^2 \boldsymbol{I}_{\boldsymbol{\theta}})$. Our unlearning criterion 1 is therefore satisfied for $\tilde{\boldsymbol{\theta}}$ (Theorem 2), and the server can perform $\tilde{N}$ new optimization rounds with FEDAVG initialized on $\tilde{\boldsymbol{\theta}}$. Thanks to the contribution of the remaining clients in $\tilde{\boldsymbol{\theta}}$, we expect the retraining with IFU to be generally faster than retraining with a random initial model.

Since $\Psi(n, i)$ is strictly increasing with n, the server can stop from computing the bounded sensitivity (8) for client i whenever $\Psi(n_i, i) > \Psi^*$ is verified after n_i optimization rounds. At this point, the model $\boldsymbol{\theta}^{n_i - 1}$ will be selected for the unlearning request of client i, as the models at subsequent iterations do not comply with the desired unlearning budget Ψ^*.

Algorithm 3 Sequential IFU (SIFU)

DURING LEARNING WITH FEDAVG

FEDAVG(I, N) initialized on initial model $\boldsymbol{\theta}_0^0$.
Compute $\Psi_0(n, i)$, equation (15).

UNLEARNING A SERIES OF REQUESTS $\{W_r\}$

Require: $\{W_r\}_{r=1}^R$, ϵ, δ, σ, and $\{N_r\}_{r=1}^R$
$O(0) = \{\emptyset\}$.
Get Ψ^* with equation (13).
for r from 1 to R **do**
$I_r = I_{r-1} \setminus W_r$.
Compute (ζ_r, T_r) with $O(r-1)$, eq. (17) and (18).
Update $O(r)$ with ζ_r, T_r, and $O(r-1)$, eq. (19).
The new global model is $\boldsymbol{\theta}_r^0 = \boldsymbol{\theta}_{\zeta_r}^{T_r} + N(0, \sigma^2 \boldsymbol{I}_{\boldsymbol{\theta}})$.
Perform FEDAVG(D_r, N_r) initialized on $\boldsymbol{\theta}_r^0$.
Compute $\Psi_r(n, i)$, eq. (15).
end for

4 Sequential FU with SIFU

In this section, we extend IFU to the sequential unlearning setting with Sequential IFU (SIFU). With Algorithm 3, SIFU is designed to satisfy a series of R unlearning requests $\{W_r\}_{r=1}^R$, where W_r is the set of clients to unlearn at request index r. SIFU generalizes IFU for which $R = 1$ and $W_1 = \{c\}$. We provide an illustration of SIFU with an example in Figure 1.

The notations introduced thus far need to be generalized to account for our series of unlearning requests $W_1, W_2, \ldots, W_R$. Global models are now referenced by their coordinates (r, n), i.e. $\boldsymbol{\theta}_r^n$, which represent the unlearning request index r and the amount of server aggregations n performed during the retraining. Hence, $\boldsymbol{\theta}_r^0$ is the initialization of the model when unlearning the clients in W_r. Also, we consider that the retraining at request index r requires N_r server aggregations on the remaining clients. Therefore, by construction, $\boldsymbol{\theta}_r^{N_r}$ is the model obtained after using SIFU to (ϵ, δ)-unlearn the sequence of unlearning requests $\{W_s\}_{s=1}^r$. Finally, we define I_r as the set of remaining clients after unlearning request r, i.e. $I_r \coloneqq I \setminus \cup_{s=1}^r W_s = I_{r-1} \setminus W_r$ with $I_0 = I$.

We extend the bounded sensitivity (8) with $\Psi_r(n, i)$ to compute the metric of client i at unlearning index r with

$$\Psi_r(n, i) \coloneqq \sum_{s=0}^{n-1} \left\| \Delta(I_r, \boldsymbol{\theta}_r^s) - \Delta(I_r \setminus \{i\}, \boldsymbol{\theta}_r^s) \right\|_2. \tag{15}$$

When unlearning client c at $r = 1$, the metric at $r = 0$ is equivalent to the previous definition of Ψ. Also, when computing the metric on a client already unlearned, i.e. $i \notin I_r$, we retrieve $\Psi_r(n, i) = 0$. Finally, for a set of clients S, we generalize the bounded sensitivity (15) to

$$\Psi_r(n, S) = \max_{i \in S} \Psi_r(n, i). \tag{16}$$

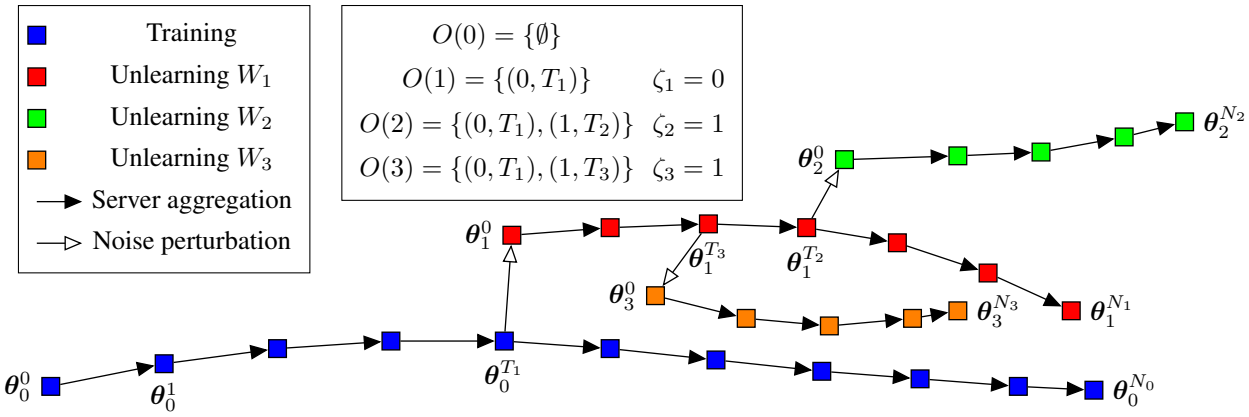

Figure 1: Illustration of SIFU (Algorithm 3) when the server receives $R = 3$ unlearning requests, through the evolution of the global model parameters $\boldsymbol{\theta}_r^n$ after server aggregation and noise perturbation. After standard federated training via FEDAVG(I, N_0), the oracle is $O(0) = \{\emptyset\}$, and the current training history is $(\boldsymbol{\theta}_0^0, \dots, \boldsymbol{\theta}_0^{N_0})$. At request $r = 1$ the unlearning index is T_1, and the training history becomes $(\boldsymbol{\theta}_0^0, \dots, \boldsymbol{\theta}_0^{T_1}, \boldsymbol{\theta}_1^0, \dots, \boldsymbol{\theta}_1^{N_1})$. The oracle is updated to $O(1) = \{(0, T_1)\}$, and $\zeta_1 = 0$. At request $r = 2$ the unlearning index is T_2 and the training history becomes $(\boldsymbol{\theta}_0^0, \dots, \boldsymbol{\theta}_0^{T_1}, \boldsymbol{\theta}_1^0, \dots, \boldsymbol{\theta}_1^{T_2}, \boldsymbol{\theta}_2^0, \dots \boldsymbol{\theta}_2^{N_2})$. The new node is added to the oracle $O(2) = \{(0, T_1), (1, T_2)\}$, and $\zeta_2 = 1$. Finally, at request $r = 3$, the unlearning index is found at $T_3 < T_2$ in the branch of request $r = 1$. The updated training history is now $(\boldsymbol{\theta}_0^0, \dots, \boldsymbol{\theta}_0^{T_1}, \boldsymbol{\theta}_1^0, \dots, \boldsymbol{\theta}_1^{T_3}, \boldsymbol{\theta}_3^0, \dots \boldsymbol{\theta}_3^{N_3})$, the oracle is updated as $O(3) = \{(0, T_1), (1, T_3)\}$, and $\zeta_3 = 1$.

With SIFU, the selection of the unlearning index T for a request r depends of the past history of unlearning requests. To keep track of the unlearning history, we introduce the oracle $O(r)$ which returns at each request r the coordinates of the history of global models where unlearning has been applied. These coordinates represent the nodes of the training history across unlearning requests (Figure 1). With reference to Figure 1, we start with the original sequence of global models obtained at each FL round, i.e. $(\boldsymbol{\theta}_0^0, \dots, \boldsymbol{\theta}_0^{N_0})$. Similarly to IFU, the first unlearning request requires to identify the unlearning index T_1 for which the corresponding global model $\boldsymbol{\theta}_0^{T_1}$ must be perturbed to obtain $\boldsymbol{\theta}_1^0$ and retrained until convergence, i.e. up to $\boldsymbol{\theta}_1^{N_1}$. The oracle is updated with the coordinates of the branching $O(1) = \{(0, T_1)\}$, and the current training history is now $(\boldsymbol{\theta}_0^0, \dots, \boldsymbol{\theta}_0^{T_1}, \boldsymbol{\theta}_1^0, \dots, \boldsymbol{\theta}_1^{N_1})$. At the next unlearning request, the server needs to identify the coordinates (ζ_r, T_r) in the new training history for which unlearning must be applied on the model $\boldsymbol{\theta}_{\zeta_r}^{T_r}$ to obtain $\boldsymbol{\theta}_r^0 = \boldsymbol{\theta}_{\zeta_r}^{T_r} + \mathcal{N}(0, \sigma^2 \boldsymbol{I}_{\boldsymbol{\theta}})$. The oracle is subsequently updated with the new set of nodes describing the new branching in the training history. By construction, we have $\zeta_r \leq r - 1$ and $T_r \leq N_{\zeta_r}$.

More precisely, we define the index ζ_r associated to the first coordinate in $O(r-1)$ for which the bounded sensitivity (15) of clients in W_r exceeds Ψ^*. Formally, we have

$$\zeta_r \coloneqq \min_s \{s : \Psi_s(n, W_r) > \Psi^* \text{ and } (s, n) \in O(r-1), \; r - 1\}. \tag{17}$$

The definition of T_r follows directly from the one of ζ_r. Similarly as for IFU, the unlearning index T_r quantifies the maximum amount of server aggregations starting from the unlearning request index ζ_r such that the bounded sensitivity $\Psi_{\zeta_r}(n, W_r)$ on this global model is inferior to Ψ^*, i.e.

$$T_r \coloneqq \arg\max_n \{\Psi_{\zeta_r}(n, W_r) \leq \Psi^*\}. \tag{18}$$

Finally, we update the oracle $O(r-1)$ to $O(r)$ with the following recurrent equation

$$O(r) = \{(s, n) \in O(r-1) \text{ s.t. } s < \zeta_r, (\zeta_r, T_r)\}. \tag{19}$$

Theorem 3 shows that for a model trained with SIFU after a given training request r, (ϵ, δ)-unlearning is guaranteed for every client belonging to the sets W_s, $s \leq r$.

Theorem 3. *The model $\boldsymbol{\theta}_r^{N_r}$ obtained with SIFU satisfies (ϵ, δ)-unlearning for every client in current and previous unlearning requests, i.e. clients in $\cup_{s=1}^r W_s$.*

Proof. See Appendix C. □

5 Experiments

In this section, we experimentally demonstrate the effectiveness of SIFU on a series of benchmarks introduced in Section 5.1. In Section 5.2, we illustrate and discuss our experimental results. Results and related code are publicly available at URL.

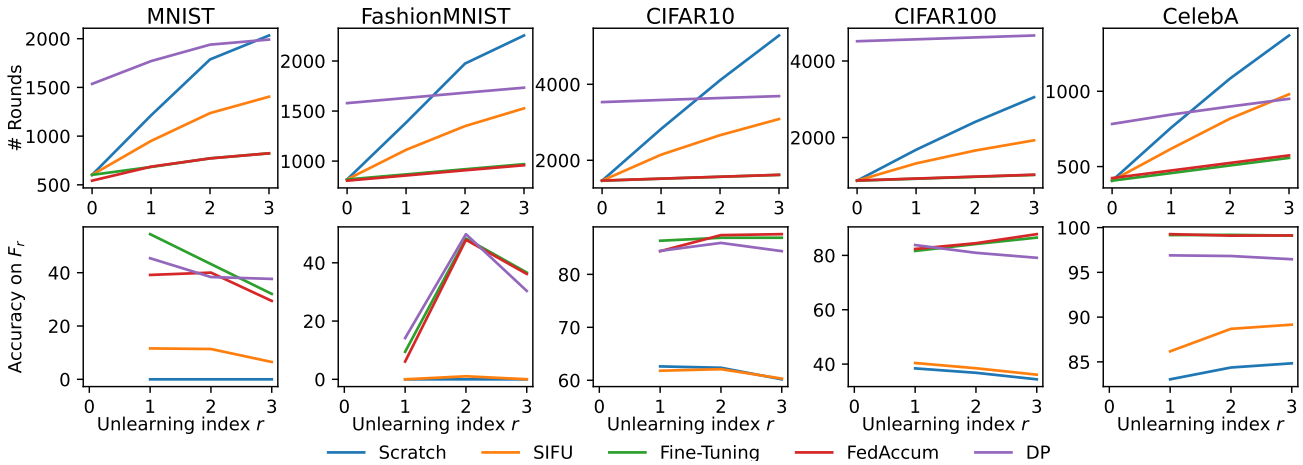

Figure 2: Total amount of aggregation rounds (1st row) and model accuracy of unlearned clients (2nd row) for MNIST, FashionMNIST, CIFAR10, CIFAR100, and CelebA (the lower the better). The server runs a federated routine with $M = 100$ clients, and unlearns 10 of them at each unlearning request ($R = 3$).

5.1 Experimental Setup

Datasets. We report experiments on reduced versions of MNIST (LeCun et al., 1998), FashionMNIST (Xiao et al., 2017), CIFAR-10 (Krizhevsky, 2009), CIFAR-100 (Krizhevsky, 2009), and CelebA (Liu et al., 2015). For each dataset, we consider $M = 100$ clients, with 100 data points each. For MNIST and FashionMNIST, each client has data samples from only one class, so that each class is represented in 10 clients only. For CIFAR10 and CIFAR100, each client has data samples with ratio sampled from a Dirichlet distribution with parameter 0.1 (Harry Hsu et al., 2019). Finally, in CelebA, clients own data samples representing the same celebrity. With these five datasets, we consider different level of heterogeneity based on label and feature distribution.

Models. For MNIST, we train a logistic regression model to consider a convex classification problem, while, for the other datasets, we train a neural network with convolutional layers followed by fully connected ones. More details on the networks are available in Appendix D.

Unlearning schemes. In addition to SIFU, we consider the following unlearning schemes from the state-of-the-art: SCRATCH, where retraining of a new initial model is performed on the remaining clients; FINE-TUNING, where retraining is performed on the current global model with the remaining clients; LAST (Neel et al., 2021), where retraining is performed on the remaining clients via perturbation of the final FL global model; DP (Dwork and Roth, 2014), where training with every client is performed with differential privacy, and FEDACCUM (Liu et al., 2021), where retraining is performed on the current global model from which the server removes the updates of the clients to unlearn, by re-aggregating the parameter updates of clients that were stored by the server across FL iterations. We provide in Appendix D the pseudo-code of FEDACCUM with the notation of our paper. We remind that FEDACCUM does not provide quantitative guarantees of the unlearning procedure, and requires the server to store the full sequence of models during the FL procedure.

Experimental scenario. We consider a sequential unlearning scenario in which the server performs the FL training procedure and then receives $R = 3$ sequential unlearning requests to unlearn 10 random clients per request. In the special case of MNIST and FashionMNIST, the server must unlearn 10 clients owning the same class. The server orchestrates each unlearning scheme through retraining until the global model accuracy on the remaining clients exceeds a fixed value specific to each dataset. We set the minimum number of 50 aggregation rounds, and a maximum budget of 10000 rounds when the stopping accuracy criterion is not met. Each unlearning method is applied with the same hyperparameters, i.e. stopping accuracy, local learning rate η, and amount of local work K (Appendix D). We define the set of clients requesting unlearning as:

$$F_r = \cup_{s=1}^{r} W_s. \tag{20}$$

In our experimental scenario, we have $|F_0| = 0$ during training and $|F_1| = 10$, $|F_2| = 20$, and $|F_3| = 30$ after each unlearning request.

Unlearning quantification. We verify the success of an unlearning scheme with two metrics: (a) the amount of server aggregation rounds needed for retraining, and (b) the resulting model accuracy on the unlearned clients. we note that, by construction, SCRATCH perfectly unlearns the clients from a request W_r. Therefore, we consider an unlearning scheme successful if it reaches similar accuracy of

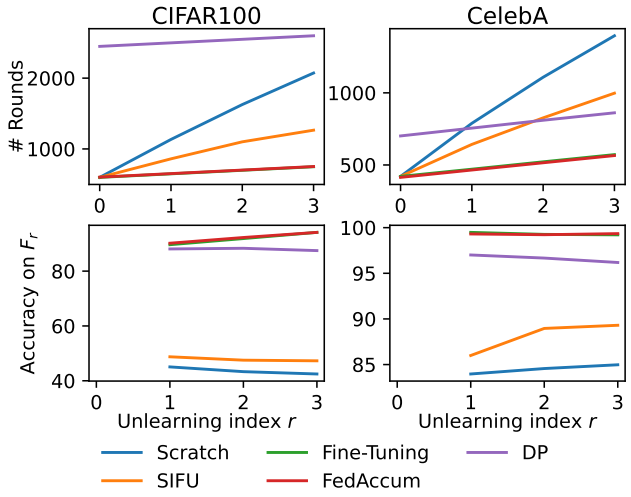

Figure 3: Total amount of aggregation rounds (1st row) and model accuracy of unlearned clients (2nd row) for the unlearning of watermarked data from CIFAR100 and CelebA.

SCRATCH with less aggregation rounds, when tested on the data samples of F_r.

5.2 Experimental Results

Figure 2 shows that for every dataset and unlearning index, FINE-TUNING, FEDACCUM, and DP provide similar model accuracy for the unlearned clients in F_r (Figure 2-2nd row), albeit significantly higher than for SCRATCH, the unlearning standard. Noteworthy, unlearning with FINE-TUNING, FEDACCUM, and DP results in significantly less aggregation rounds than SCRATCH (Figure 2-1st row). We note that SIFU and SCRATCH lead to similar unlearning results, quantified by low accuracy on the unlearned clients F_r (Figure 2-2nd row), while SIFU unlearns these clients in roughly half the amount of aggregation rounds needed for SCRATCH (Figure 2-1st row). However, the model accuracy of SIFU is slightly higher than the one of SCRATCH, with perfect overlap only for FashionMNIST. This behavior is natural and can be explained by our privacy budget (ϵ, δ), which trades unlearning capabilities for effectiveness of the retraining procedure. With highest unlearning budget, i.e. $\epsilon = 0$ and $\delta = 0$, SIFU would require to retrain from the initial model $\boldsymbol{\theta}_0^0$, thus reducing to SCRATCH.

Finally, we observed that when unlearning with LAST, the retrained model always converged to a local optimum with accuracy inferior to our target after 10000 aggregation rounds. This behavior is likely due to the difficulty of calibrating the noise perturbation due to the numerous heterogeneous contributions of the clients. For this reason, we decided to exclude LAST from the plots of Figure 2.

5.3 Verifying Unlearning through Watermarking

The work of Sommer et al. (2020) proposes an adversarial approach to verify the efficiency of an unlearning scheme based on watermarking. We apply here this method to our federated setting, in which watermarking is operated by each client by randomly assigning on all its data samples the maximum possible value to 10 given pixels. To ensure that clients' data heterogeneity is only due to the modification of the pixels, we define heterogeneous data partitioning across clients by randomly assigning the data according to a Dirichlet distribution with parameter 1. Figure 3 shows our results for this experimental scenario on CIFAR100 and CelebA, while Appendix D provides similar results for MNIST, FashionMNIST and CIFAR10. We retrieve the same conclusions drawn from Figure 2. SIFU and SCRATCH have similar accuracies on the unlearned clients in F_r, to demonstrate the effectiveness of the unlearning. Moreover, SIFU unlearns these clients in significantly less aggregation rounds than SCRATCH.

5.4 Impact of the noise perturbation on SIFU

Appendix D illustrates the impact of the perturbation amplitude σ on convergence speed when unlearning with SIFU. We note that when unlearning with a small σ, SIFU has identical behavior to SCRATCH as the unlearning is applied to the initial random model $\boldsymbol{\theta}_0^0$. With large values of σ, SIFU performs instead identically to LAST and applies the unlearning to the finale global model $\boldsymbol{\theta}_r^{N_r}$.

6 Conclusions

In this work, we introduce informed federated unlearning (IFU), a novel federated unlearning scheme to unlearn a client's contribution from a model trained with federated learning. Upon receiving an unlearning request from a given client, IFU identifies the optimal FL iteration from which FL has to be reinitialised, with statistical unlearning guarantees defined by Definition 1. We extend the theory of IFU to account for the practical scenario of sequential unlearning (SIFU), where the server receives a series of forgetting request of one or more clients. We prove that SIFU can unlearn a series of forgetting requests while satisfying our unlearning guarantees, and demonstrate the effectiveness of our methods on a variety of tasks and dataset.

An additional contribution of this work consists in a new theory for bounding the clients contribution in FL. The server can compute this bound for every client without asking for any additional computation and communication. The theoretical justification of this approach relies on the linear approximation of the clients' loss function, and its relevance is here demonstrated across several benchmarks. Future extensions of the work will focus on generalizing our unlearning framework to more general settings.

Acknowledgments

This work has been supported by the French government, through the 3IA Côte d'Azur Investments in the Future project managed by the National Research Agency (ANR) with the reference number ANR-19-P3IA-0002, and by the ANR JCJC project Fed-BioMed 19-CE45-0006-01. The project was also supported by Accenture. The authors are grateful to the OPAL infrastructure from Université Côte d'Azur for providing resources and support.

References

Acar, D. A. E., Zhao, Y., Matas, R., Mattina, M., Whatmough, P., and Saligrama, V. (2021). Federated learning based on dynamic regularization. In *International Conference on Learning Representations*.

Basu, D., Data, D., Karakus, C., and Diggavi, S. (2019). Qsparse-local-sgd: Distributed sgd with quantization, sparsification and local computations. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Bourtoule, L., Chandrasekaran, V., Choquette-Choo, C. A., Jia, H., Travers, A., Zhang, B., Lie, D., and Papernot, N. (2021). Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 141–159. IEEE.

Cao, Y. and Yang, J. (2015). Towards making systems forget with machine unlearning. *2015 IEEE Symposium on Security and Privacy*, pages 463–480.

Chen, X., Wu, S. Z., and Hong, M. (2020). Understanding gradient clipping in private sgd: A geometric perspective. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 13773–13782. Curran Associates, Inc.

Dwork, C. and Roth, A. (2014). The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3–4):211–407.

Fraboni, Y., Vidal, R., Kameni, L., and Lorenzi, M. (2021a). Clustered sampling: Low-variance and improved representativity for clients selection in federated learning. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 3407–3416. PMLR.

Fraboni, Y., Vidal, R., Kameni, L., and Lorenzi, M. (2021b). On the impact of client sampling on federated learning convergence. *CoRR*, abs/2107.12211.

Fredrikson, M., Jha, S., and Ristenpart, T. (2015). Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, page 1322–1333, New York, NY, USA. Association for Computing Machinery.

Ginart, A., Guan, M., Valiant, G., and Zou, J. Y. (2019). Making ai forget you: Data deletion in machine learning. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Golatkar, A., Achille, A., Ravichandran, A., Polito, M., and Soatto, S. (2021). Mixed-privacy forgetting in deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 792–801.

Golatkar, A., Achille, A., and Soatto, S. (2020a). Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Golatkar, A., Achille, A., and Soatto, S. (2020b). Forgetting outside the box: Scrubbing deep networks of information accessible from input-output observations.

Guo, C., Goldstein, T., Hannun, A., and Van Der Maaten, L. (2020). Certified data removal from machine learning models. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3832–3842. PMLR.

Gupta, V., Jung, C., Neel, S., Roth, A., Sharifi-Malvajerdi, S., and Waites, C. (2021). Adaptive machine unlearning. In Ranzato, M., Beygelzimer, A., Nguyen, K., Liang, P. S., Vaughan, J. W., and Dauphin, Y., editors, *Advances in Neural Information Processing Systems*, volume 34, pages 16319–16330. Curran Associates, Inc.

Haddadpour, F., Kamani, M. M., Mahdavi, M., and Cadambe, V. (2019). Trading redundancy for communication: Speeding up distributed SGD for non-convex optimization. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2545–2554. PMLR.

Halimi, A., Kadhe, S., Rawat, A., and Baracaldo, N. (2022). Federated unlearning: How to efficiently erase a client in fl? *arXiv preprint arXiv:2207.05521*.

Harding, E. L., Vanto, J. J., Clark, R., Hannah Ji, L., and Ainsworth, S. C. (2019). Understanding the scope and impact of the california consumer privacy act of 2018. *Journal of Data Protection & Privacy*, 2(3):234–253.

Harry Hsu, T. M., Qi, H., and Brown, M. (2019). Measuring the effects of non-identical data distribution for federated visual classification. *arXiv*.

Izzo, Z., Anne Smart, M., Chaudhuri, K., and Zou, J. (2021). Approximate data deletion from machine learning models. In Banerjee, A. and Fukumizu, K., editors,

Proceedings of The 24th International Conference on Artificial Intelligence and Statistics, volume 130 of *Proceedings of Machine Learning Research*, pages 2008–2016. PMLR.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *ICLR (Poster)*.

Krizhevsky, A. (2009). Learning multiple layers of features from tiny images.

LeCun, Y., Bottou, L., Bengio, Y., and Ha, P. (1998). LeNet. *Proceedings of the IEEE*, (November):1–46.

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. (2018). Federated Optimization in Heterogeneous Networks. *Proceedings of the 1 st Adaptive & Multitask Learning Workshop, Long Beach, California, 2019*, pages 1–28.

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smithy, V. (2019). Feddane: A federated newton-type method. In *2019 53rd Asilomar Conference on Signals, Systems, and Computers*, pages 1227–1231.

Li, X., Huang, K., Yang, W., Wang, S., and Zhang, Z. (2020). On the convergence of fedavg on non-iid data. In *International Conference on Learning Representations*.

Li, X. and Orabona, F. (2019). On the convergence of stochastic gradient descent with adaptive stepsizes. In Chaudhuri, K. and Sugiyama, M., editors, *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 983–992. PMLR.

Liu, G., Ma, X., Yang, Y., Wang, C., and Liu, J. (2021). 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.

Liu, Z., Luo, P., Wang, X., and Tang, X. (2015). Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.

Mahadevan, A. and Mathioudakis, M. (2021a). Certifiable machine unlearning for linear models. *CoRR*, abs/2106.15093.

Mahadevan, A. and Mathioudakis, M. (2021b). Certifiable machine unlearning for linear models. *arXiv preprint arXiv:2106.15093*.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. (2017). Communication-Efficient Learning of Deep Networks from Decentralized Data. In Singh, A. and Zhu, J., editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282, Fort Lauderdale, FL, USA. PMLR.

Neel, S., Roth, A., and Sharifi-Malvajerdi, S. (2021). Descent-to-delete: Gradient-based methods for machine unlearning. In Feldman, V., Ligett, K., and Sabato, S., editors, *Proceedings of the 32nd International Conference on Algorithmic Learning Theory*, volume 132 of *Proceedings of Machine Learning Research*, pages 931–962. PMLR.

Reisizadeh, A., Mokhtari, A., Hassani, H., Jadbabaie, A., and Pedarsani, R. (2020). Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization. In Chiappa, S. and Calandra, R., editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 2021–2031. PMLR.

Sattler, F., Müller, K.-R., and Samek, W. (2021). Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE Transactions on Neural Networks and Learning Systems*, 32(8):3710–3722.

Sommer, D. M., Song, L., Wagh, S., and Mittal, P. (2020). Towards probabilistic verification of machine unlearning. *CoRR*, abs/2003.04247.

Voigt, P. and Von dem Bussche, A. (2017). The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 10(3152676):10–5555.

Wang, H., Sievert, S., Liu, S., Charles, Z., Papailiopoulos, D., and Wright, S. (2018). Atomo: Communication-efficient learning via atomic sparsification. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Wang, J., Guo, S., Xie, X., and Qi, H. (2021). Federated unlearning via class-discriminative pruning.

Wang, J., Liu, Q., Liang, H., Joshi, G., and Poor, H. V. (2020). Tackling the objective inconsistency problem in heterogeneous federated optimization. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Ward, R., Wu, X., and Bottou, L. (2019). AdaGrad stepsizes: Sharp convergence over nonconvex landscapes. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6677–6686. PMLR.

Wu, C., Zhu, S., and Mitra, P. (2022). Federated unlearning with knowledge distillation. *arXiv preprint arXiv:2201.09441*.

Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747.

Yu, H., Jin, R., and Yang, S. (2019a). On the linear speedup analysis of communication efficient momentum SGD for distributed non-convex optimization. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7184–7193. PMLR.

Yu, H., Yang, S., and Zhu, S. (2019b). Parallel Restarted SGD with Faster Convergence and Less Communication: Demystifying Why Model Averaging Works for Deep Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):5693–5700.

A When fine tuning does not guarantee unlearning: example on linear regression

Let us consider a linear regression optimization, with feature matrix $\boldsymbol{X}$ and predictions $\boldsymbol{y}$ such that the loss function f is defined as

$$f(\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{\theta}) = \frac{1}{2} \left[\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}\right]^T \left[\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}\right]. \tag{21}$$

In this example, we assume there are more features than data samples, which makes $\boldsymbol{X}^T\boldsymbol{X}$ a singular matrix. While f is convex, f has more than one global optimum. Any model with parameter $\boldsymbol{\theta}^*$ such that

$$\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\theta}^* = \boldsymbol{X}^T\boldsymbol{y} \tag{22}$$

is a global optimum. When $\boldsymbol{X}^T\boldsymbol{X}$ is non-singular, we retrieve the unique optimum in close-form $\boldsymbol{\theta}^* = \left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{y}$. We show with this simple example that, upon unlearning a data sample, no amount of fine-tuning on the model $\boldsymbol{\theta}^*$ can lead to the same model obtained when retraining from a random initial model. We differentiate between $(\boldsymbol{X}, \boldsymbol{y})$ and $(\boldsymbol{X}_{-1}, \boldsymbol{y}_{-1})$ our data with and without a given data point.

Optimizing f, as defined in equation (21), with N steps of gradient descent, learning rate η, and initial model $\boldsymbol{\theta}_0$ gives model parameters $\boldsymbol{\theta}^N$ defined as

$$\boldsymbol{\theta}^N = \underbrace{\left[I - \eta\boldsymbol{X}^T\boldsymbol{X}\right]^N}_{A(\boldsymbol{X},N)}\boldsymbol{\theta}^0 + \underbrace{\eta\sum_{n=0}^{N-1}\left[I - \eta\boldsymbol{X}^T\boldsymbol{X}\right]^n\boldsymbol{X}^T\boldsymbol{y}}_{B(\boldsymbol{X},\boldsymbol{y},N)}. \tag{23}$$

We first note that we retrieve the standard form for the global optimum of linear regression when $\boldsymbol{X}^T\boldsymbol{X}$ is non-singular as $\lim_{n\to\infty} A(\boldsymbol{X}, n) = 0$ and $\lim_{n\to\infty} B(\boldsymbol{X}, \boldsymbol{y}, n) = \left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{y}$. In the general form accounting for the singular case, at least one eigenvalue of $A(\boldsymbol{X}, N)$ is equal to 1 independently from the amount of gradient descent steps N. Hence, the parameters of the model obtained with gradient descent optimization always depend from the ones of the initial model $\boldsymbol{\theta}^0$. Hence, when unlearning our data sample from $\boldsymbol{\theta}^N$, the resulting trained model still depends of that data samples. Indeed, if we compare the model $\boldsymbol{\theta}^{\tilde{N}}_{-1}$ trained on the data samples $(\boldsymbol{X}_{-1}, \boldsymbol{y}_{-1})$, to the model $\phi^{\tilde{N}}_{-1}$ obtained after fine-tuning the model $\boldsymbol{\theta}^N$ with $\tilde{N}$ server aggregations, we have

$$\phi^{\tilde{N}}_{-1} - \boldsymbol{\theta}^{\tilde{N}}_{-1} = A(\boldsymbol{X}_{-1}, \tilde{N})A(\boldsymbol{X}, N)\boldsymbol{\theta}^0 + A(\boldsymbol{X}_{-1}, \tilde{N})B(\boldsymbol{X}, \boldsymbol{y}, N). \tag{24}$$

B Forgetting a Single Client with IFU, Proof of Theorem 1

We first consider the case where clients perform $K = 1$ SGD in Section B.1 before considering the general case $K \geq 1$ in Section B.2.

B.1 Proof of Theorem 1 for $K = 1$

Proof. We define by $\boldsymbol{\theta}^N = \textsc{FedAvg}(I, N)$ and $\boldsymbol{\phi}^N = \textsc{FedAvg}(I_{-c}, N)$ the models trained with FedAvg on $\boldsymbol{\theta}_0$ with respectively all the clients, i.e. I, and all the clients but client c, i.e. I_{-c}, performing $K = 1$ GD step.

When clients perform $K = 1$ GD step, two consecutive global models can be related, when training with clients in I as a simple GD step, i.e.

$$\boldsymbol{\theta}^{n+1} = \boldsymbol{\theta}^n - \eta\nabla f_I(\boldsymbol{\theta}^n). \tag{25}$$

By considering the same process for I_{-c} and with Assumption 1, we get

$$\begin{aligned}\boldsymbol{\phi}^{n+1} - \boldsymbol{\theta}^{n+1} &= \boldsymbol{\phi}^n - \boldsymbol{\theta}^n - \eta\left[\nabla f_{I_{-c}}(\boldsymbol{\phi}^n) - \nabla f_I(\boldsymbol{\theta}^n)\right] &(26)\\ &= \left[I - \eta H_{I_{-c}}(\boldsymbol{\theta}^n)\right]\left[\boldsymbol{\phi}^n - \boldsymbol{\theta}^n\right] - \eta\left[\nabla f_{I_{-c}}(\boldsymbol{\theta}^n) - \nabla f_I(\boldsymbol{\theta}^n)\right]. &(27)\end{aligned}$$

$H_{I_{-c}}(\boldsymbol{\theta}^n)$ is semi-positive, Assumption 1. Let us define $\sigma_{\max}(H_{I_{-c}}(\boldsymbol{\theta}^n))$ the highest eigenvalue of $H_{I_{-c}}(\boldsymbol{\theta}^n)$. When consider that $\eta \leq 1/\sigma_{\max}(H_{I_{-c}}(\boldsymbol{\theta}^n))$, and due to the subadditivity of the norm, we get the following recurrent inequality

$$\left\|\boldsymbol{\phi}^{n+1} - \boldsymbol{\theta}^{n+1}\right\|_2 \leq \eta\left\|\nabla f_I(\boldsymbol{\theta}^n) - \nabla f_{I_{-c}}(\boldsymbol{\theta}^n)\right\|_2 + \left\|\boldsymbol{\phi}^n - \boldsymbol{\theta}^n\right\|_2, \tag{28}$$

which when developed completes the proof when clients perform $K = 1$ GD. □

B.2 Proof of Theorem 1 for $K \geq 1$

Proof. We maintain the definitions of $\boldsymbol{\theta}^n$ and $\boldsymbol{\phi}^n$ introduced in Section B.1. To account for the amount of local work K, we introduce $\boldsymbol{\theta}_i^{n,k}$ the model of client i after k GD steps performed on global model $\boldsymbol{\theta}^n$. We apply a similar reasoning for $\boldsymbol{\phi}_i^{n,k}$.

With Assumption 1, we have

$$\nabla f_i(\boldsymbol{\phi}_i^{n,k}) = \nabla f_i(\boldsymbol{\theta}_i^{n,k}) + H_i(\boldsymbol{\theta}_i^{n,k}) \left(\boldsymbol{\phi}_i^{n,k} - \boldsymbol{\theta}_i^{n,k}\right), \tag{29}$$

which gives

$$\begin{aligned}
\boldsymbol{\phi}_i^{n,k+1} - \boldsymbol{\theta}_i^{n,k+1} &= \left(\boldsymbol{\phi}_i^{n,k+1} - \boldsymbol{\phi}_i^{n,k}\right) - \left(\boldsymbol{\theta}_i^{n,k+1} - \boldsymbol{\theta}_i^{n,k}\right) + \left(\boldsymbol{\phi}_i^{n,k} - \boldsymbol{\theta}_i^{n,k}\right) && (30)\\
&= -\eta \left[\nabla f_i\left(\boldsymbol{\phi}_i^{n,k}\right) - \nabla f_i\left(\boldsymbol{\theta}_i^{n,k}\right)\right] + \left(\boldsymbol{\phi}_i^{n,k} - \boldsymbol{\theta}_i^{n,k}\right) && (31)\\
&= \left[I - \eta H_i(\boldsymbol{\theta}_i^{n,k})\right] \left(\boldsymbol{\phi}_i^{n,k} - \boldsymbol{\theta}_i^{n,k}\right) && (32)\\
&= \left[\prod_{r=0}^{k} \left[I - \eta H_i(\boldsymbol{\theta}_i^{n,r})\right]\right] (\boldsymbol{\phi}^n - \boldsymbol{\theta}^n), && (33)
\end{aligned}$$

where the third equality follows from equation (29), and the fourth from expanding the recurrent equation. For the rest of this work, we define $Q_i^n = \prod_{k=0}^{K-1} \left[I - \eta H_i(\boldsymbol{\theta}_i^{n,k})\right]$.

Using equation (33), we relate the difference between two global models with every client in I and in I_c. When removing client c the clients' importance changes. We consider importance p_i when training with I. Instead, when training with clients in I_c, we consider the regularized importance $q_i = p_i/(1-p_c)$ for the remaining clients and $q_c = 0$. We have

$$\begin{aligned}
\boldsymbol{\phi}^{n+1} - \boldsymbol{\theta}^{n+1} &= \sum_{i=1}^{M} q_i \left(\boldsymbol{\phi}_i^{n+1} - \boldsymbol{\phi}^n\right) - \sum_{i=1}^{M} p_i \left(\boldsymbol{\theta}_i^{n+1} - \boldsymbol{\theta}^n\right) && (34)\\
&= \sum_{i=1}^{M} q_i \left[\left(\boldsymbol{\phi}_i^{n+1} - \boldsymbol{\theta}_i^{n+1}\right) + \left(\boldsymbol{\theta}_i^{n+1} - \boldsymbol{\theta}^n\right)\right] - \sum_{i=1}^{M} p_i \left(\boldsymbol{\theta}_i^{n+1} - \boldsymbol{\theta}^n\right) && (35)\\
&= \left(\sum_{i=1}^{M} q_i Q_i^n\right) (\boldsymbol{\phi}^n - \boldsymbol{\theta}^n) + \Delta(I_{-c}, \boldsymbol{\theta}^n) - \Delta(I, \boldsymbol{\theta}^n). && (36)
\end{aligned}$$

We consider a learning rate η such that $\eta \leq 1/\sigma_{\max}(H_i(\boldsymbol{\theta}^{n,k}))$. Hence, $\|Q_i^n\|_2 \leq 1$. With equation (36), we get the following inequality

$$\left\|\boldsymbol{\phi}^{n+1} - \boldsymbol{\theta}^{n+1}\right\|_2 \leq \|\boldsymbol{\phi}^n - \boldsymbol{\theta}^n\|_2 + \|\Delta(I, \boldsymbol{\theta}^n) - \Delta(I_{-c}, \boldsymbol{\theta}^n)\|_2, \tag{37}$$

which expansion completes the proof.

□

B.3 Local Loss Functions' Regularization and Strong Convexity, Proof of Corollary 1

Proof. Under L2 regularization, every client's regularized loss function F_i is expressed as

$$F_i(\boldsymbol{\theta}) = f_i(\boldsymbol{\theta}) + \frac{\lambda}{2} \|\boldsymbol{\theta}\|^2 \text{ and } \nabla F_i(\boldsymbol{\theta}) = \nabla f_i(\boldsymbol{\theta}) + \lambda \boldsymbol{\theta}. \tag{38}$$

When clients perform $K = 1$ GD step, equation (36) reduces to

$$\boldsymbol{\phi}^{n+1} - \boldsymbol{\theta}^{n+1} = \eta \left[\nabla f_I(\boldsymbol{\theta}^n) - \nabla f_{I_{-c}}(\boldsymbol{\theta}^n)\right] + \left[(1-\eta\lambda)I - \eta H_{I_{-c}}(\theta^n)\right] (\boldsymbol{\phi}^n - \theta^n), \tag{39}$$

which, if $\eta \leq 1/(\lambda + \sigma_{\max}(H_i(\boldsymbol{\theta}^n))$, gives

$$\left\|\boldsymbol{\phi}^{n+1} - \boldsymbol{\theta}^{n+1}\right\|_2 \leq \eta \left\|\nabla f_I(\boldsymbol{\theta}^n) - \nabla f_{I_{-c}}(\boldsymbol{\theta}^n)\right\|_2 + (1 - \eta\lambda - \eta\mu) \left\|\boldsymbol{\phi}^n - \boldsymbol{\theta}^n\right\|_2 . \tag{40}$$

When clients perform $K \geq 1$ GD steps, we have $\boldsymbol{\phi}_i^{n+1} - \boldsymbol{\theta}_i^{n+1} = Q_i^n \left[\boldsymbol{\phi}^n - \boldsymbol{\theta}^n\right]$ with

$$Q_i^n = \prod_{r=0}^{K-1} \left[(1 - \eta\lambda)I - \eta H_i(\boldsymbol{\theta}_i^{n,r})\right] . \tag{41}$$

Hence, we retrieve equation (36). We consider the local learning rate satisfy $\eta \leq 1/(\lambda + \sigma_{\max}(H_i(\boldsymbol{\theta}^n)))$. Hence, considering that Q_i^n can be bounded with the μ-strong convexity of the Hessian, we get

$$\left\|\boldsymbol{\phi}^{n+1} - \boldsymbol{\theta}^{n+1}\right\|_2 \leq \eta \left\|\Delta(I, \boldsymbol{\theta}^n) - \Delta(I_{-c}, \boldsymbol{\theta}^n)\right\|_2 + (1 - \eta\lambda - \eta\mu)^K \left\|\boldsymbol{\phi}^n - \boldsymbol{\theta}^n\right\|_2 . \tag{42}$$

Developing this recurrent equation completes the proof.

□

B.4 Generalization

The proof of Theorem 1 can be also extended to account for FL regularization methods (Li et al., 2018, 2019; Acar et al., 2021), other SGD solvers (Kingma and Ba, 2015; Ward et al., 2019; Li and Orabona, 2019; Yu et al., 2019a,b; Haddadpour et al., 2019), client sampling (Li et al., 2018, 2020; Fraboni et al., 2021b) and/or gradient compression/quantization (Reisizadeh et al., 2020; Basu et al., 2019; Wang et al., 2018).

B.5 Calculus simplification with uniform importance

For computation purposes, we propose the following expression to estimate a client bounded sensitivity, equation (8. When removing client c, each client has new importance $q_i = p_i/(1 - p_c)$ for the remaining clients and $q_c = 0$. Hence, we have

$$\left\|\Delta(I, \boldsymbol{\theta}^n) - \Delta(\boldsymbol{\theta}^n, \mathcal{D}_{-c})\right\|_2 = \left\| \left[\boldsymbol{\theta}^{n+1} - \boldsymbol{\theta}^n\right] - \left[\sum_{i=1}^M q_i \boldsymbol{\theta}_i^{n+1} - \boldsymbol{\theta}^n\right] \right\|_2 \tag{43}$$

$$= \left\| \boldsymbol{\theta}^{n+1} - \frac{1}{1 - p_c} \left[\boldsymbol{\theta}^{n+1} - p_c \boldsymbol{\theta}_i^{n+1}\right] \right\|_2 \tag{44}$$

$$= \frac{p_c}{1 - p_c} \left\|\boldsymbol{\theta}_i^{n+1} - \boldsymbol{\theta}^{n+1}\right\|_2 \tag{45}$$

In the special case where clients have identical importance, we have $p_c/(1 - p_c) = 1/(M - 1)$.

C Convergence of SIFU, Theorem 3

C.1 Intermediate results

Property 1. *If there exists ν, s, u such that $s < u$, $(\nu, t_s) \in O(s)$ and $(\nu, t_u) \in O(u)$, then $t_s \geq t_u$.*

Proof. We first assume that s and u satisfy $u = s + 1$. Considering that $(\nu, t_s) \in O(s)$ and $(\nu, t_u) \in O(u)$, we have, by definition of ζ_u in equation (17), $\nu \leq \zeta_u$.

- $\zeta_u > \nu$. Considering that $u = s + 1$, we have $t_s = t_u$, equation (19).
- $\zeta_u = \nu$. Considering that $(\nu, t_s) \in O(s)$ and $(\nu, t_u) \in O(u)$, then we have $\nu \leq s - 1$. Therefore, by definition of ζ_u, we have $\Psi_{\zeta_u}(t_s, W_u) > \Psi^*$. By construction of T_u, equation (18), we have $t_u = T_u < t_s$.

When considering the more general case where there exists an integer k such that $u = s + k$ while $(\nu, t_s) \in O(s)$ and $(\nu, t_u) \in O(u)$, then it is sufficient to consider iteratively an integer j ranging from 1 to k. Considering $(\nu, t_u) \in O(u)$, there exists t_{s+j} such that $(\nu, t_{s+j}) \in O(s + j)$. In that case, using the same reasoning as for $k = 1$, we have $t_s \leq t_{s+1} \leq \ldots \leq t_{s+k-1} \leq t_u$.

□

C.2 Proof of Theorem 3

Proof. Proving that $\boldsymbol{\theta}_r^{N_r}$ (ϵ, δ)-unlearns every client in F_r, equation (20), reduces to proving that $\boldsymbol{\theta}_r^0$ (ϵ, δ)-unlearns every client in F_r, equation (20). Indeed, the data of clients in F_r are not used on the noised perturbed model $\boldsymbol{\theta}_r^0 = \boldsymbol{\theta}_{\zeta_r}^{T_r} + \mathcal{N}(0, \sigma^2 \boldsymbol{I}_{\boldsymbol{\theta}})$.

We prove by induction that $\boldsymbol{\theta}_r^0$ (ϵ, δ)-unlearns every client in F_r, equation (20). The initialization ($r = 1$) directly follows from IFU, Algorithm 2, with Theorem 2. We now assume that for every s such that $s \leq r - 1$, $\boldsymbol{\theta}_s^0$ (ϵ, δ)-unlearns every client in F_s and prove that $\boldsymbol{\theta}_r^0$ (ϵ, δ)-unlearns every client in F_r.

- $\underline{s \leq \zeta_r}$. Using the induction property, $\boldsymbol{\theta}_{\zeta_r}^0$ (ϵ, δ)-unlearns every clients in W_s. Clients in W_s are not used for training on $\boldsymbol{\theta}_{\zeta_r}^0$. Hence, $\boldsymbol{\theta}_{\zeta_r}^{T_r}$ and $\boldsymbol{\theta}_r^0$ also (ϵ, δ)-unlearns every client in W_s.
- $\underline{s = r.}$ By definition of ζ_r, equation (17), the noise perturbations for every model in $O(r)$ is such that $\boldsymbol{\theta}_{\zeta_r}^0$ (ϵ, δ)-unlearns every client in W_r. Hence, by definition of T_r on the bounded sensitivity of clients in W_r at unlearning request ζ_r, equation (18), the noised perturbed model $\boldsymbol{\theta}_r^0$ (ϵ, δ)-unlearns every client in W_r, Theorem 2.
- $\underline{\zeta_r < s \leq r - 1.}$ The successive update of the oracle, equation (19), from $O(\zeta_r)$ to $O(s)$ gives, by construction, that there exists t_s such that the coordinates (ζ_r, t_s) are in $O(s)$. Hence, by definition of ζ_s, equation (17), we have $\zeta_s \geq \zeta_r$ and the successive noise perturbations to obtain $\boldsymbol{\theta}_{\zeta_r}^0$ (ϵ, δ)-unlearns every client in W_s. Also, while we have the coordinates (ζ_r, t_s) in $O(s)$, we also have the coordinates (ζ_r, T_r) in $O(r)$, equation (19). Therefore, using property 1, we have $t_s \geq T_r$. Hence, we have $\Psi_{\zeta_r}(T_r, W_s) \leq \Psi^*$. Therefore, with the noise perturbation of SIFU, clients in W_s are (ϵ, δ)-unlearned in $\boldsymbol{\theta}_r^0$.

□

D Experiments

For every benchmark, we consider the number of SGD steps K, batch size B, number of clients M, the number of sampled clients m, the standard deviation σ of the noise perturbation, and the local learning rate η given in Table 1. Also, for our unlearning scheme SIFU, DP, and LAST, we consider an unlearning budget of $\epsilon = 10$ and $\delta = 0.01$. The unlearning budget plays the important role of identifying in the training history the global model to perturb. Theorem 2 shows that ϵ and σ are linearly related. Hence, to unlearn a client c from a global model c, a smaller σ can be considered, but at the cost of a lower unlearning budget (ϵ, δ), Definition 1. Also, for fair comparison of DP with other FU schemes, we select the best clipping value C, in a range from 0.001 to 1, for which the global model reaches the target accuracy in the smallest amount of aggregation rounds. Finally, for FashionMNIST, CIFAR10, CIFAR100, and CelebA, we consider model architectures composed of three convolutional layers followed by two fully connected layers, with implementation at URL.

Table 1: Hyperparameters used for our different unlearning benchmarks described in Section 5.1.

Dataset	K	B	M	m	σ	η	C
MNIST	10	100	100	10	0.05	0.01	0.5
FashionMNIST	5	20	100	10	0.1	0.02	0.5
CIFAR10	5	20	100	5	0.05	0.01	0.2
CIFAR100	5	20	100	5	0.05	0.02	0.2
CelebA	10	20	100	20	0.1	0.01	0.5

The training and retraining depends on the initial model $\boldsymbol{\theta}_0^0$ and the clients' batches of data used at every aggregation to compute their local work. Hence, we replicate each unlearning scenario on 10 different seeds and plot in Figure 2 to 6 their averaged results. For the unlearning benchmarks described in Section 5.1 and used in Figure 2, 5, and 6, the stopping accuracies considered are 93% for MNIST, 90% for FashionMNIST, CIFAR10, and CIFAR100, and 99.9% for CelebA. For Figure 3 and 4 with unlearning benchmark described in Section 5.3, the stopping accuracies considered are instead 99.9% for MNIST, FashionMNIST, CIFAR10, and CelebA, and 99% for CIFAR100. Reaching such accuracies is easier with the backdoored datasets because the clients' data heterogeneity is only due to their watermark, Section 5.3.

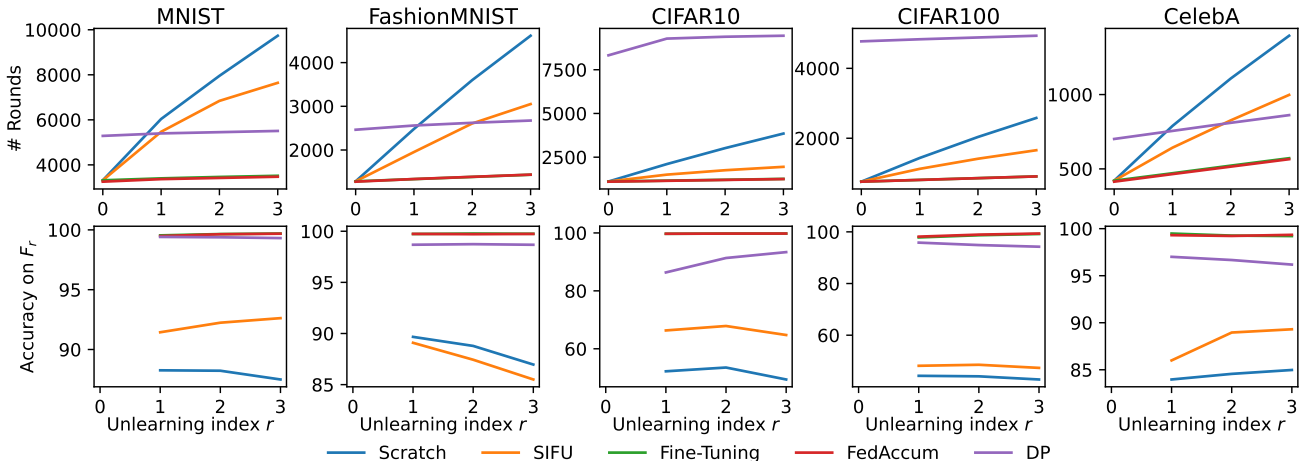

Figure 4: Total amount of aggregation rounds (1st row) and model accuracy of unlearned clients (2nd row) for the unlearning of watermarked data from MNIST, FashionMNIST, CIFAR10, CIFAR100, and CelebA (the lower the better).

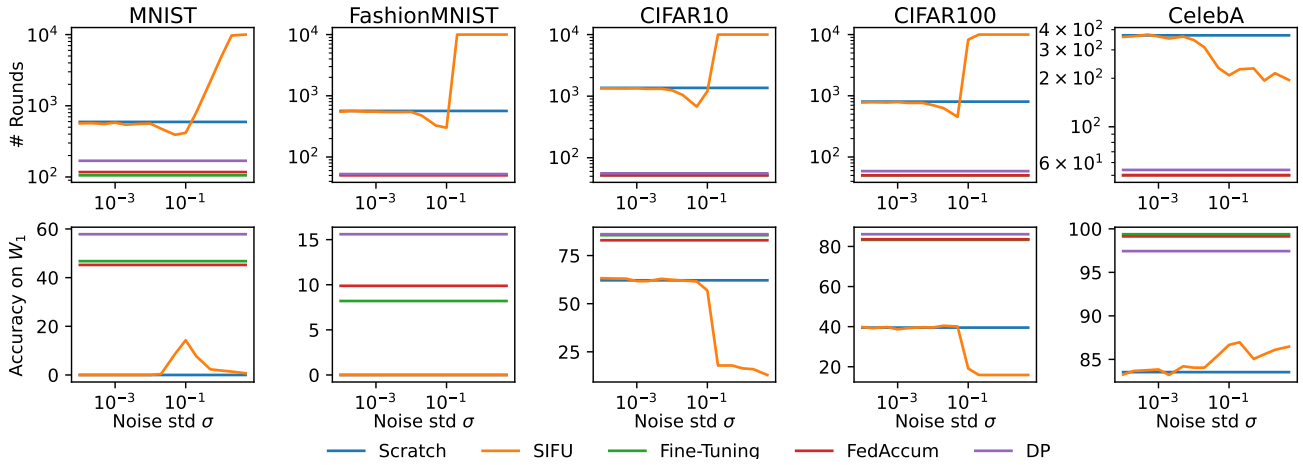

Figure 5: Impact of the noise standard deviation σ when unlearning with SIFU versus SCRATCH. Total amount of aggregation rounds (1st row) and model accuracy of unlearned clients (2nd row) for MNIST, FashionMNIST, CIFAR10, CIFAR100, and CelebA (the lower the better).

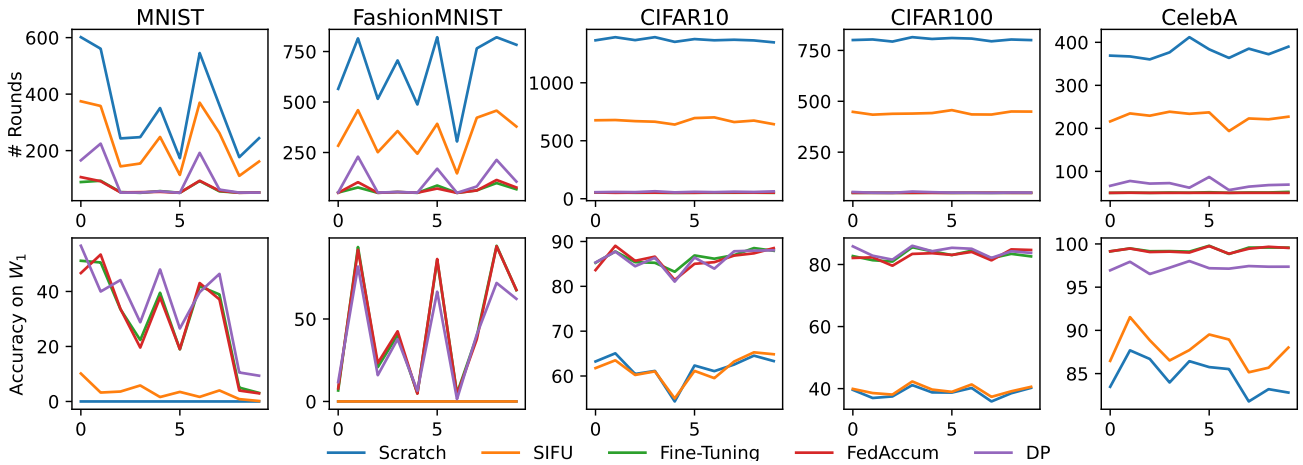

Figure 6: Total amount of aggregation rounds (1st row) and model accuracy of unlearned clients (2nd row) for MNIST, FashionMNIST, CIFAR10, CIFAR100, and CelebA (the lower the better). This figure displays the unlearning capabilities of the unlearning benchmarks introduced in Section 5.1 after training on clients in I and unlearning $|W_1| = 10$ clients. For each integer on the x-axis, a different set of clients to unlearn is considered. Each unlearning request is composed of 10 random clients for CIFAR10, CIFAR100, and CelebA. For MNIST and FashionMNIST, each unlearning request $|W_1|$ has 10 clients of the same class such that the x-axis is the class forgotten. The integers on the x-axis corresponds to the class of the clients to unlearn.