Fully Decentralized Certified Unlearning

Hithem Lamri Michail Maniatakos
Center for Cyber Security, New York University Abu Dhabi
Abu Dhabi, United Arab Emirates
{hithem.lamri,michail.maniatakos}@nyu.edu

Abstract

*Machine unlearning (MU) seeks to remove the influence of specified data from a trained model in response to privacy requests or data poisoning. While certified unlearning has been analyzed in centralized and server-orchestrated federated settings (via guarantees analogous to differential privacy, DP), the decentralized setting—where peers communicate without a coordinator—remains underexplored. We study certified unlearning in decentralized networks with fixed topologies and propose **RR-DU**, a random-walk procedure that performs one projected gradient ascent step on the forget set at the unlearning client and a geometrically distributed number of projected descent steps on the retained data elsewhere, combined with subsampled Gaussian noise and projection onto a trust region around the original model. We provide (i) convergence guarantees in the convex case and stationarity guarantees in the non-convex case, (ii) (ε, δ) network-unlearning certificates on client views via subsampled Gaussian Rényi DP (RDP) with segment-level subsampling, and (iii) deletion-capacity bounds that scale with the forget-to-local data ratio and quantify the effect of decentralization (network mixing and randomized subsampling) on the privacy–utility trade-off. Empirically, on image benchmarks, **RR-DU**matches a given (ε, δ) while achieving higher test accuracy than decentralized DP baselines and reducing backdoor accuracy (ASR) to the random-guessing baseline ($\approx 10\%$).*

1. Introduction

Machine Unlearning (MU) aims to remove the influence of a designated subset—the unlearning set—from a trained model while preserving performance on the retained data. As ML is deployed in healthcare, finance, and vision, regulations such as GDPR [18] and CCPA [40] formalize a *right to be forgotten*, and the growing risk of data poisoning further motivates MU. Simply deleting records is insufficient—their effect persists in the model parameters. The naive remedy, retraining from scratch without the unlearning set, is often impractical for large models and production systems. This raises the core challenge: efficiently erasing a subset's influence while maintaining utility.

Machine Unlearning was first introduced by Cao and Yang [10], and the field has since grown into two main categories: *exact unlearning* (retraining from scratch) and *approximate unlearning* (reducing the cost of retraining while matching the from-scratch reference up to a tolerance). MU was later extended to Federated Learning as Federated Unlearning (FU) [35], with heavy study in both server-orchestrated [36] and fully decentralized scenarios. The decentralized nature of FL has led many works to adopt relaxed assumptions, and obtaining rigorous certified unlearning guarantees [4, 43] is difficult; results often hold only under (strongly) convex objectives, while many server-based and decentralized FU methods [2, 24, 34, 46, 49–51] are heuristic and lack theoretical guarantees.

Zhang et al. [53] provided Differential Privacy (DP) [16]-based unlearning guarantees for server-orchestrated FL via perturbed improved retraining where all clients and the server collaborate, which is computationally expensive by design. On the other hand, Qiao et al. [42] gave the first certified unlearning for fully decentralized unlearning under dynamic topologies, introducing two algorithms that involve all clients or all neighbors and require storing past gradients; this induces high storage/compute overhead and temporal test-accuracy drops during unlearning. In addition, that work does not articulate decentralized *view*-specific issues in certification, lacks a deletion-capacity analysis [4, 43, 48] that formalizes the utility–unlearning trade-off, and does not separate DP-based guarantees from certified unlearning as done in centralized settings [4, 26, 43]. Altogether, certified decentralized unlearning on fixed graphs remains underexplored and needs a rigorous reformulation.

Following this line of work, and inspired by privacy amplification from decentralization [14] and subsampling [5, 19], we address limitations in certified decentralized unlearning. We ask: can unlearning be performed *autonomously* without involving all clients on a fixed topol-

ogy; what are the effects of decentralization (routing, mixing, trust regions) on the utility–privacy trade-off and on deletion capacity; and is Decentralized DP (DDP) still a good candidate for certified unlearning? To answer these questions, we reformulate certified unlearning in decentralized settings where each client has only a partial view of the network; we then introduce a new approximate decentralized certified unlearning algorithm on a fixed topology that combines projected noisy gradient ascent on the unlearning client only with randomized client sampling, called **RR-DU** (randomized-restart decentralized unlearning), while other clients continue regular training. We incorporate the relevant amplification analyses (network mixing, projection, subsampling) [5, 14, 19], show why DDP is not an ideal unlearning certifier due to noise scaling with the forget-set size m, and prove that **RR-DU** attains stronger guarantees with substantially less noise (not scaling with m), together with unlearning–utility and deletion-capacity analyses for both settings; we also provide convergence guarantees for strongly convex, convex, and smooth nonconvex losses.

We support our claims with experiments on *three real-world datasets* and *two model architectures*, showing that RR-DU reduces backdoor accuracy (ASR) to the retraining baseline (near random guessing) while maintaining higher clean accuracy, demonstrating that it is practical with low communication and storage overhead.

Our **contributions** can be summarized as follows:

- *Formulation & capacity (first to our knowledge).* We introduce a *view-based* formulation of certified decentralized unlearning on fixed graphs and adapt deletion capacity to decentralized settings, making explicit the roles of routing probabilities, network mixing, and trust-region projections.
- *Algorithm.* We propose **RR-DU**: a lightweight token method that performs projected noisy ascent on the forget set *only at the unlearning client*, uses randomized routing with optional per-node averaging, and requires neither storing past gradients nor involving all clients.
- *Theory.* We derive (ε, δ) decentralized certificates on client *views* via RDP with subsampling and network amplification; prove last-iterate/stationarity guarantees for strongly convex, convex, and smooth nonconvex losses; and give a two-regime deletion-capacity characterization that separates optimization/variance from an alignment-bias term—clarifying when **RR-DU** outperforms DDP. This also yields a clean separation between DDP-as-certifier (group-privacy scales with the number of deletions) and approximate certified unlearning, for which **RR-DU**'s noise does not scale with the forget-set size.
- *Empirical evaluation and code availability.* Under matched privacy budgets, RR-DU reduces backdoor accuracy (ASR) to near the random-guessing baseline while preserving higher clean accuracy than DDP and fine-tuning baselines. The implementation is publicly available at this repository.

2. Related Work

Private Decentralized Learning Motivated by privacy and scalability concerns [20, 45], a large body of work studies *fully decentralized* FL algorithms that avoid a central coordinator and rely instead on peer-to-peer exchanges along edges of a network graph—either under dynamic topologies [28, 32, 33] or fixed topologies [39]. Decentralized stochastic gradient methods (often called incremental/gossip methods) operate via gossip protocols [9, 13]. To satisfy privacy requirements, Differential Privacy (DP)—originally developed in the centralized model (DP-SGD [1])—has been adapted to decentralized settings. In particular, Cyffers and Bellet [14] introduced *network differential privacy* to relax the stronger local DP model [15, 27], leveraging privacy amplification by subsampling [5], by shuffling [6, 7, 17], and by iteration/contractive maps (e.g., Euclidean projection) [19], together with Rényi DP [37], to obtain *amplification by decentralization* on fixed topologies. Recent work further tightens convergence and privacy via matrix factorization views [8], correlated noise [3], and synthetic-data amplification [41]. While DP is a strong notion that implies certified unlearning (CU) through group privacy and composition [47], our focus here is to adopt the perspective of [14] and show *why decentralized DP (DDP) is not an ideal certifier for unlearning*, despite being theoretically valid. We make a clean separation between DDP and decentralized certified unlearning (DCU).

Certified Unlearning There is a growing literature on CU in centralized settings: many results target convex objectives and linear models [21, 23, 43, 48], while others extend to non-convex tasks [4, 11, 12, 38] and neural networks [29, 52]. In fully decentralized FL, the only work studying CU under random topologies and gossip was introduced by Qiao et al. [42], which uses the Gaussian mechanism but involves all users (or all neighbors) and stores past gradients—incurring significant storage/compute overhead and a temporary drop in test accuracy during unlearning. Moreover, it imports CU definitions from the centralized setting and certifies *per client*, without extending guarantees to network-level *views*. In contrast, we add noise *only at the unlearning client* and perform *noisy projected gradient ascent* there, while others continue standard training; randomized next-hop selection and per-hop step counts act as *post-processing*, amplifying privacy and yielding tighter bounds without relying on DDP. Our gradient-alignment design makes the required noise *independent of the forget-set size*, improving the privacy–utility trade-off.

3. Problem Statement

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a fixed, undirected graph with $\mathcal{V} = \{1, \dots, N\}$ users; an edge $(u, v) \in \mathcal{E}$ means u can communicate with v. This formulation encompasses both complete graphs, in which every pair of users is adjacent, and non-complete connected graphs, which capture more constrained communication topologies. Each user u holds a private dataset D_u of size $n_u := |D_u|$. We denote the global dataset on the graph by $D := \bigcup_{u \in \mathcal{V}} D_u$ with total size $n := |D| = \sum_u n_u$, which is independently drawn from a distribution $\mathcal{P}$ over the data space $\mathcal{Z}$. Users want to collaboratively learn a shared model represented by a parameter vector $\theta \in \Theta \subseteq \mathbb{R}^d$. Given a loss function $\ell : \mathbb{R}^d \times \mathcal{Z} \to \mathbb{R}_+$, the goal is to minimize the population risk:

$$\mathcal{L}(\theta) := \mathbb{E}_{z \sim \mathcal{P}}[\, \ell(\theta, z)\,]. \tag{1}$$

However, during the *training* phase, we reduce the problem to empirical risk minimization:

$$\min_{\theta \in \Theta} \mathcal{L}(\theta, \mathcal{V}) := \frac{1}{|\mathcal{V}|} \sum_{u \in \mathcal{V}} \ell_u(\theta), \tag{2}$$

where ℓ_u is the local objective function of user u, defined as

$$\ell_u(\theta) := \frac{1}{n_u} \sum_{z \in D_u} \ell(\theta; z). \tag{3}$$

We denote by $\mathcal{A}$ our training algorithm. It is basically implemented as a single *token* carrying the current model θ that performs a random walk on $\mathcal{G}$ for T rounds. When user u receives the token, it runs local SGD steps on D_u and forwards the token (with updated θ) to a uniformly random neighbor. Each user updates only upon token arrival (see Algorithm **1** in Appendix B).

Moving to the *unlearning* scenario: at some round $t \in \{1, \dots, T\}$, user u receives a deletion request for a subset $D_f \subseteq D_u$ with $m := |D_f|$. Retraining from scratch on $D \setminus D_f$ is the natural approach, but it is often impractical; instead we apply an *unlearning algorithm* $\mathcal{U}$ to transform the output of the original algorithm, $\mathcal{A}(D)$, into an *unlearned* model that is distributionally close to the output of a suitable certifier that has no access to D_f. This is known as *certified unlearning*; formally defined as follows.

Definition 3.1 ((ε, δ)-Certified Unlearning)**.** Let D be a dataset of size n drawn from a distribution $\mathcal{P}$, and let $D_f \subseteq D$ be a delete set with $|D_f| \leq m$. Let $\mathcal{A}$ be a learning algorithm that outputs $\mathcal{A}(D) \in \Theta$, and let $\mathcal{U}$ be an unlearning algorithm that, given a delete (forget) set D_f, a model, and data statistics $T(D)$, outputs $\mathcal{U}(D_f, \mathcal{A}(D), T(D)) \in \Theta$. We say that $(\mathcal{A}, \mathcal{U})$ is (ε, δ)-*unlearning* if there exists a (possibly problem-dependent) *certifying algorithm* $\mathcal{C}$ such that for all measurable sets $\theta \subseteq \Theta$:

$$\mathbb{P}\big[\mathcal{U}(D_f, \mathcal{A}(D), T(D)) \in \theta\big] \leq e^{\varepsilon}\, \mathbb{P}\big[\mathcal{C}(D \setminus D_f) \in \theta\big] + \delta,$$
$$\mathbb{P}\big[\mathcal{C}(D \setminus D_f) \in \theta\big] \leq e^{\varepsilon}\, \mathbb{P}\big[\mathcal{U}(D_f, \mathcal{A}(D), T(D)) \in \theta\big] + \delta.$$

Thus, the output distribution after unlearning is (ε, δ)-indistinguishable from that of a certifying procedure that has no access to the forget set D_f. Typical choices include $\mathcal{C}(D \setminus D_f) = \mathcal{A}(D \setminus D_f)$ Ginart et al. [21], Guo et al. [23] or $\mathcal{C}(D \setminus D_f) = \mathcal{U}(\varnothing, \mathcal{A}(D \setminus D_f), T(D \setminus D_f))$ Allouah et al. [4], Sekhari et al. [43]. In this work, we adopt the second choice, noting that this is just a theoretical choice and does not affect the results in any way.

Decentralized Formulation. We adopt decentralized differential privacy (DDP) of Cyffers and Bellet [14]. A decentralized algorithm $\mathcal{A}$ on a graph produces a transcript $\mathcal{A}(D)$ of all exchanged messages. No user sees the full transcript: user u only observes a *view*

$$O_u(\mathcal{A}(D)) \;=\; \{\, (v, m, v') \in \mathcal{A}(D) \;:\; v = u \text{ or } v' = u \,\}. \tag{4}$$

For each v, let $\Theta_v := \mathrm{Range}(O_v)$ denote the observation space of v's views.

Definition 3.2 (Network Differential Privacy [14])**.** An algorithm $\mathcal{A}$ satisfies (ε, δ)-network DP if for all distinct $u, v \in \mathcal{V}$, all neighboring datasets $D \sim_u D'$ (differing only in user u's data), and all $\theta \subseteq \Theta_v$,

$$\mathbb{P}\big[\, O_v(\mathcal{A}(D)) \in \theta \,\big] \;\leq\; e^{\varepsilon}\, \mathbb{P}\big[\, O_v(\mathcal{A}(D')) \in \theta \,\big] + \delta. \tag{5}$$

We adapt certified unlearning to this view-based setting:

Definition 3.3 ((ε, δ)-Decentralized Certified Unlearning)**.** Let $\mathcal{A}$ produce $\mathcal{A}(D)$ and let $\mathcal{U}$ produce $\mathcal{U}(D_f, \mathcal{A}(D))$. We say $(\mathcal{A}, \mathcal{U})$ achieves (ε, δ) *decentralized certified unlearning* if there exists a certifier $\mathcal{C}$ with transcript $\mathcal{C}(D \setminus D_f)$ such that for any deletion request by user u (i.e., $D_f \subseteq D_u$), any $v \neq u$, and all $\theta \subseteq \Theta_v$,

$$\mathbb{P}\big[\, O_v(\mathcal{U}(D_f, \mathcal{A}(D))) \in \theta \,\big] \;\leq\; e^{\varepsilon}\, \mathbb{P}\big[\, O_v(\mathcal{C}(D \setminus D_f)) \in \theta \,\big] + \delta, \tag{6}$$

and *symmetrically* with $\mathcal{U}$ and $\mathcal{C}$ swapped.

We note that our guarantee is *view-based* for any non-deleting client $v \neq u$ and is explicitly *post-deletion*: it certifies the distribution of the observer view $\mathcal{O}_v$ after the deletion request and during unlearning, but it cannot retroactively erase information already revealed in pre-deletion training transcripts.

To quantify the trade-off between the statistical guarantee of Definition 3.1 and the resulting model's utility, we adopt the *Deletion Capacity* metric from Sekhari et al. [43].

Definition 3.4 (Deletion capacity)**.** Let $\varepsilon, \delta \geq 0$. Let $D \sim \mathcal{P}^n$ be drawn i.i.d. from a distribution $\mathcal{P}$, and let $\ell(\theta, z)$ be a loss. Define the population risk $\mathcal{L}(\theta) = \mathbb{E}_{z\sim\mathcal{P}}[\,\ell(\theta, z)\,]$ and $\mathcal{L}^* = \min_{\theta\in\Theta} \mathcal{L}(\theta)$. For a pair $(\mathcal{A},\mathcal{U})$ that is (ε, δ)-unlearning (per Definition 3.1 or Definition 3.3), the *deletion capacity* $m_{\varepsilon,\delta}^{\mathcal{A},\mathcal{U}}(d, N)$ is the largest integer m such that

$$\mathbb{E}\left[\max_{D_f\subseteq D:\,|D_f|\leq m} \left(\mathcal{L}(\mathcal{U}(D_f, \mathcal{A}(D), T(D))) - \mathcal{L}^*\right)\right] \leq \gamma,$$

where the expectation is taken over $D \sim \mathcal{P}^n$ and over the internal randomness of $\mathcal{A}$ and $\mathcal{U}$ (and any randomness in T).

Unlearning via Differential Privacy. The view-based definitions let us certify unlearning by privacy: Differential Privacy (DP) implies global certified unlearning (Definition 3.1) with certifier $\mathcal{C}(D \setminus D_f) = \mathcal{A}(D \setminus D_f)$, and Network-DP (Definition 3.2) implies decentralized certified unlearning on *views* (Definition 3.3). Thus any network-private token algorithm (see Algorithm **2** in Appendix B) can serve as its own certifier by simply running the private protocol: $\mathcal{U}(D_f, \mathcal{A}(D)) = \mathcal{A}(D)$ and comparing to $\mathcal{C}(D\setminus D_f) = \mathcal{A}(D\setminus D_f)$. We now quantify the deletion capacity (Definition 3.4) achievable under this decentralized-DP baseline.

Theorem 3.5 (Deletion capacity of decentralized DP)**.** *Let $\Theta \subset \mathbb{R}^d$ be convex with diameter $R := \sup_{\theta,\theta'\in\Theta} \|\theta-\theta'\|_2$, and assume the loss ℓ is L-smooth and convex. Consider the network-private SGD (Appendix B), run for T hops with N clients, and take $\mathcal{U}(D_f, \mathcal{A}(D), T(D)) = \mathcal{A}(D)$. Then, for target (ε, δ) at edit distance m, the deletion capacity satisfies*

$$m_{\varepsilon,\delta}^{\mathcal{A},\mathcal{U}}(d, N) \;=\; \widetilde{\Omega}\left(\frac{\varepsilon}{R\,L\,(2+\log T)}\,\sqrt{\frac{N}{d\,\log(1/\delta)\,\log N}}\right). \tag{7}$$

Proof sketch. We combine utility, decentralized view calibration, and group privacy, then solve for m. The full proof is deferred to Appendix C.1.

Decentralization Effect on Deletion Capacity. For the DDP baseline [14], Theorem 3.5 shows that the view-based calibration yields the lower bound in (7). In words, capacity *increases* with the number of users N and *decreases* with the model dimension d, failure probability δ, and the optimization horizon via the $(2 + \log T)$ factor. Operationally, T tracks the total number of stochastic updates observed across the network (each user contributes about T/N in expectation). By contrast, in centralized DP, classical analyses give analogous deletion-capacity guarantees where the key driver is the total dataset size n rather than the number of nodes N [4, 26, 43]. Full details are deferred to Appendix C.2.

If each random-walk message aggregates $s \geq 1$ independent unbiased gradients with independent Gaussian noise terms *before* any observer's first view, the effective variance becomes $G^2 = L^2 + d\sigma^2/s$, which improves capacity by a factor $\sqrt{s}$. The corresponding scaling is

$$m_{\varepsilon,\delta}^{\mathcal{A},\mathcal{U}}(d, N) \;=\; \widetilde{\Omega}\left(\frac{\varepsilon}{R\,L\,(2+\log T)}\,\sqrt{\frac{s\,N}{d\,\log(1/\delta)\,\log N}}\right). \tag{8}$$

Two special cases recover common protocols: (i) *one-update-per-hop* random walk (s=1) reduces to (7); (ii) synchronous rounds that average over all users ($s\approx N$) yield an extra $\sqrt{N}$ gain relative to one-update-per-hop.

4. Algorithm Design

4.1. Motivation and Problem Setup

As discussed in Section 3, Decentralized Differential Privacy (DDP) [14] can deliver decentralized certified unlearning (DCU), but it typically injects noise at *every* step, which degrades utility. In our random-walk setting, the forget set $D_f \subseteq D_u$ affects the computation only when the walker is at the unlearning user u; updates performed at other users are independent of D_f and thus constitute *post-processing* with respect to D_f. This observation motivates our design: add Gaussian noise *only at* u, keep other users noiseless, and exploit network mixing so that any given observer sees only a limited fraction of sensitive events. The result is a DCU mechanism with markedly smaller effective variance than network-wide DDP under the same (ε, δ) view-privacy budget.

4.2. Mathematical Foundation: Gradient Alignment

We work with the user-averaged empirical risk

$$\mathcal{L}(\theta, \mathcal{V}) := \frac{1}{N}\sum_{v\in\mathcal{V}} \ell_v(\theta), \quad \ell_v(\theta) := \frac{1}{n_v}\sum_{z\in D_v} \ell(\theta; z). \tag{9}$$

After deleting $D_f \subseteq D_u$ of size m, the retraining objective becomes

$$\begin{aligned} \mathcal{L}_{\setminus f}(\theta, \mathcal{V}) &:= \frac{1}{N}\left(\ell_{u\setminus f}(\theta) + \sum_{v\neq u} \ell_v(\theta)\right), \\ \ell_{u\setminus f}(\theta) &:= \frac{1}{n_u - m}\sum_{z\in D_u\setminus D_f} \ell(\theta; z). \end{aligned} \tag{10}$$

Let $\ell_f(\theta) := \frac{1}{m}\sum_{z\in D_f}\ell(\theta;z)$ and assume L-smoothness ($\|\nabla\ell(\theta;z)\|_2 \le L$). A direct calculation gives

$$\nabla\mathcal{L}_{\setminus f}(\theta,\mathcal{V}) = \nabla\mathcal{L}(\theta,\mathcal{V}) - \frac{1}{N}\Big[\nabla\ell_u(\theta) - \nabla\ell_{u\setminus f}(\theta)\Big] + \Delta_{\mathrm{norm}}(\theta). \tag{11}$$

where $\|\Delta_{\mathrm{norm}}(\theta)\|_2 = O\Big(L\,\frac{m}{n_u}\Big)$. Thus, moving toward the retrained optimum can be realized by standard descent on $v \neq u$ plus a *corrective step at* u that depends only on local information.

Decentralized Realization via Random Walk. We realize **RR-DU** through the randomized-restart procedure in Algorithm 1. At each step, the token returns to the deleting client u with probability p; otherwise, it moves according to the routing transition matrix P of the communication graph $\mathcal{G}$ (uniform on a complete graph). Formally, for any connected graph, $v_{t+1} = u$ with probability p, and $v_{t+1} \sim P(v_t,\cdot)$ otherwise. Only client u performs the noisy corrective update, while all other clients take noiseless PGD steps and may average $s \ge 1$ mini-batches to reduce variance. In both the theoretical analysis and the main experiments, we focus on complete graphs for simplicity. Choosing $p = 1/N$ matches the visit rate of a uniform random walk and yields clean alignment weights, while the trust-region projection $\Pi_{\mathbb{B}(\theta_{\mathrm{ref}},\varrho)}$ stabilizes the noisy ascent at u.

Exact vs. Lightweight Alignment at u When the walker is at u, we use either *exact alignment* ($-\nabla\ell_{u\setminus f}(\theta)$) or *lightweight alignment* $\big(\frac{m}{n_u}\nabla\ell_f(\theta)\big)$. Both fit the view-based privacy accounting; the lightweight choice trades a controlled alignment bias for lower compute/memory.

4.3. Synthesis of Advantages

Let $g_{\neg u}(\theta) := \frac{1}{N-1}\sum_{v\neq u}\nabla\ell_v(\theta)$. The conditional expected update satisfies

$$\frac{\mathbb{E}[\Delta\theta_t \mid \theta_t]}{\eta_t} = -(1-p)\,g_{\neg u}(\theta_t) + p\,g_u(\theta_t). \tag{12}$$

With $p = \frac{1}{N}$ and the *exact-alignment* choice $g_u(\theta) = -\nabla\ell_{u\setminus f}(\theta)$, we recover the retraining direction:

$$\frac{\mathbb{E}[\Delta\theta_t \mid \theta_t]}{\eta_t} = -\frac{1}{N}\left(\sum_{v\neq u}\nabla\ell_v(\theta_t) + \nabla\ell_{u\setminus f}(\theta_t)\right) = -\nabla\mathcal{L}_{\setminus f}(\theta_t,\mathcal{V}). \tag{13}$$

With the *lightweight* choice $g_u(\theta) = \frac{m}{n_u}\nabla\ell_f(\theta)$, the alignment error is controlled:

$$\left\|\tfrac{\mathbb{E}[\Delta\theta_t|\theta_t]}{\eta_t} + \nabla\mathcal{L}_{\setminus f}(\theta_t,\mathcal{V})\right\|_2 = O\Big(L\,\tfrac{m}{n_u}\Big). \tag{14}$$

Algorithm 1 RR-DU: Randomized-Restart Decentralized Unlearning

Require: Initial θ_0; unlearning user u; forget set $D_f \subseteq D_u$; routing prob. p; stepsizes $\{\eta_t\}$; noise scale σ; feasible set Θ; trust ball $\mathbb{B}(\theta_{\mathrm{ref}},\varrho)$; horizon T_u; local-averaging $s \ge 1$; **mode** $\in \{\textsc{exact}, \textsc{lightweight}\}$

$\theta \leftarrow \theta_0$
for $t = 1$ **to** T_u **do**
  Route: with prob. p move to u; else to $v_{t+1} \sim P(v_t,\cdot)$ ▷ P is the transition matrix of the graph $\mathcal{G}$
  if current node is u **then** ▷ *Noisy* corrective step at u
    draw $Z_t \sim \mathcal{N}(0,\sigma^2 I_d)$
    if mode $=$ EXACT **then**
      $g_u \leftarrow -\nabla\ell_{u\setminus f}(\theta)$
    else ▷ LIGHTWEIGHT
      $g_u \leftarrow \frac{m}{n_u}\nabla\ell_f(\theta)$
    end if
    $\theta \leftarrow \Pi_{\mathbb{B}(\theta_{\mathrm{ref}},\varrho)}\big(\theta + \eta_t\,(g_u + Z_t)\big)$
  else ▷ *Noiseless* PGD on D_v
    draw $B_v^{(1)},\dots,B_v^{(s)} \subseteq D_v$ i.i.d.
    $g_v \leftarrow \frac{1}{s}\sum_{i=1}^{s}\nabla\ell(\theta;B_v^{(i)})$
    $\theta \leftarrow \Pi_{\Theta}\big(\theta - \eta_t\,g_v\big)$
  end if
end for
return θ

Effective Variance: Concentrated Noise and Local Averaging Only a fraction p of hops add Gaussian noise, so the per-hop second moment obeys

$$\overline{G}^2 := \frac{1}{T_u}\sum_{t=1}^{T_u}\mathbb{E}\big[\|g_t\|_2^2\big] \;\le\; L^2 \;+\; \frac{p}{s}\,d\,\sigma^2, \tag{15}$$

where local averaging s reduces stochastic variance at non-u users by $\approx 1/s$. In contrast, network-wide DDP adds $d\,\sigma^2$ on every hop.

Geometric mixing and stability. For any observer $v \neq u$, the first-observation delay of a sensitive update is $\mathrm{Geom}(q)$ with $q = \frac{1-p}{N-1}$, since the walker must leave u (w.p $1-p$) and land at v (w.p $1/(N-1)$). This mixing underpins the view-level privacy amplification used in Section 5. Moreover, trust-region projection keeps noisy steps controlled:

$$\|\theta_{t+1} - \theta_t\|_2 \;\le\; \eta_t\,\big\|g_t + Z_t\,\mathbf{1}_{\{\text{at } u\}}\big\|_2, \tag{16}$$

ensuring the iterate stays within $\mathbb{B}(\theta_{\mathrm{ref}},\varrho)$ while Π_Θ maintains feasibility elsewhere.

5. Theoretical Analysis

5.1. Assumptions and Setup

(Informal) We assume: (i) L-smooth losses and non-expansive projections (onto Θ and the trust ball $\mathbb{B}(\theta_{\mathrm{ref}},\varrho)$);

Table 1. **Utility bounds across objective classes (RR-DU).** Optimization and privacy terms shown separately; combine additively. Measure indicates the quantity being bounded.

Objective class	Optimization term	Privacy term (using Cor. 5.2)	Measure
Convex (bounded domain)	$\tilde{O}\Big(\frac{R_{\mathrm{cert}}\,L}{\sqrt{s\,T_u}}\Big)$	$\tilde{O}\Big(R_{\mathrm{cert}}\,\frac{L}{\varepsilon}\,p\,\sqrt{\frac{d\,\ln(1/\delta)\,\ln N}{s\,N}}\Big)$	$\mathcal{L}(\theta_{T_u}) - \mathcal{L}^\star$
μ-Strongly convex	$\tilde{O}\Big(\frac{L^2}{\mu\,s\,T_u}\Big)$	$\tilde{O}\Big(\frac{L^2}{\mu}\,\frac{1}{\varepsilon}\,p\,\sqrt{\frac{d\,\ln(1/\delta)\,\ln N}{s\,N}}\Big)$	$\mathcal{L}(\theta_{T_u}) - \mathcal{L}^\star$
Smooth nonconvex	$\tilde{O}\Big(\frac{L^2}{\sqrt{s\,T_u}}\Big)$	$\tilde{O}\Big(\frac{L^2}{\varepsilon}\,p\,\sqrt{\frac{d\,\ln(1/\delta)\,\ln N}{s\,N}}\Big)$	$\frac{1}{T_u}\sum_{t=1}^{T_u}\|\nabla\mathcal{L}(\theta_t)\|^2$

(ii) *localized* Gaussian noise is injected only when the random walk is at the unlearning user u and touches D_f (other updates are post-processing w.r.t. D_f); (iii) routing visits u with probability p and otherwise a uniformly sampled node in $\mathcal{V}\setminus\{u\}$ (for the complete graph case); (iv) non-u nodes may average $s \geq 1$ i.i.d. mini-batches before forwarding. Let T_u denote the number of unlearning rounds (hops) and $R_{\mathrm{cert}} := \mathrm{diam}\big(\Theta \cap \mathbb{B}(\theta_{\mathrm{ref}}, \varrho)\big)$.

We analyze convex, strongly convex, and smooth nonconvex objectives; *formal* statements and technical conditions appear in Appendix C.3.

5.2. Privacy on Views (Network-DP) and Noise Calibration

Theorem 5.1 ((ε,δ)-DCU via view-based amplification)**.** *Fix $p \in (0,1)$ and horizon T_u. In **RR-DU**, only visits to u are sensitive. Let the single-visit Gaussian mechanism at u have RDP level ε_0 with failure δ_0. Then, for any observer $v \neq u$, the composed view-privacy parameters satisfy*

$$\begin{aligned} \varepsilon &= O\Big(\varepsilon_0\,\sqrt{p\,T_u\,\tfrac{\ln N}{N}\,\ln\tfrac{1}{\delta'}}\Big), \\ \delta &= O\Big(p\,T_u\,\delta_0\,\tfrac{\ln N}{N}\Big) + \delta'. \end{aligned} \tag{17}$$

Proof sketch. Count sensitive visits $M_u \sim \mathrm{Binomial}(T_u, p)$; use geometric first-observation and weak convexity of D_α to obtain the $\sqrt{\ln N/N}$ amplification; compose in RDP and convert to (ε,δ). Full proofs are deferred to C.4.

Corollary 5.2 (Noise calibration (scaling))**.** *With $\delta' = \delta/2$ and $\delta_0 = \Theta\big(\frac{\delta\,N}{p\,T_u\,\ln N}\big)$, a Gaussian scale achieving (ε,δ) on views obeys*

$$\sigma = \Theta\left(\frac{L}{\varepsilon}\,\sqrt{\frac{p\,T_u\,\ln(1/\delta)\,\ln N}{N}}\right). \tag{18}$$

5.3. Utility: Last-Iterate Excess Risk (with averaging s)

Let $\mathcal{L}(\theta)$ denote the retraining objective on $D \setminus D_f$. On bounded domains, projected (stochastic) first-order methods decompose last-iterate guarantees into an *optimization term* plus a *variance term*. In **RR-DU**, only a fraction p of hops (those at the unlearning user u) inject Gaussian noise, while non-u hops may average $s \geq 1$ mini-batches; the per-hop second moment thus satisfies

$$G^2 \;\leq\; L^2 \;+\; \frac{p}{s}\,d\,\sigma^2. \tag{19}$$

Plugging the calibrated noise from Cor. 5.2 yields the bounds summarized in Table 1 (all $\tilde{O}$ hide logarithms in T_u). The optimization and privacy terms add. Full proofs are provided on Appendix C.5.

5.4. Alignment Bias and Deletion Capacity

Because RR-DU performs corrective ascent on D_f only at u, the retraining-vs-unlearning gap includes an *alignment bias*

$$\mathrm{bias}(m) \;=\; \Theta\Big(L\,\frac{m}{n_u}\Big), \tag{20}$$

arising from the $n_u \to n_u - m$ renormalization and imperfect first-order alignment. Putting everything together, for a target tolerance $\gamma > 0$, we require

$$\underbrace{\tilde{O}\Big(\tfrac{R_{\mathrm{cert}}\,L}{\sqrt{s\,T_u}}\Big) \;+\; \tilde{O}\Big(R_{\mathrm{cert}}\,\tfrac{L}{\varepsilon}\,p\,\sqrt{\tfrac{d\,\ln(1/\delta)\,\ln N}{s\,N}}\Big)}_{A\ \text{(non-bias term, independent of } m)} \;+\; \underbrace{C\,L\,\tfrac{m}{n_u}}_{\text{alignment bias}} \;\leq\; \gamma. \tag{21}$$

Hence the *deletion capacity* is fundamentally two-regime:

$$m^\star \;=\; \begin{cases} \Omega\Big(\frac{(\gamma - A)\,n_u}{L}\Big), & \text{if } \gamma > A, \\ 0, & \text{otherwise.} \end{cases} \tag{22}$$

Regime transition. The variance-limited regime occurs when $A \geq \gamma$, where capacity is determined by solving:

$$\tilde{O}\Big(\tfrac{R_{\mathrm{cert}}\,L}{\sqrt{s\,T_u}}\Big) + \tilde{O}\Big(R_{\mathrm{cert}}\,\tfrac{L}{\varepsilon}\,p\,\sqrt{\tfrac{d\,\ln(1/\delta)\,\ln N}{s\,N}}\Big) \;\leq\; \gamma. \tag{23}$$

The bias-limited regime occurs when $A < \gamma$, yielding

$$m \;=\; \Omega\Big(\tfrac{(\gamma - A)\,n_u}{L}\Big). \tag{24}$$

Table 2. **DDP vs. RR-DU (convex case, corrected).** DDP (group privacy) requires noise that scales linearly with m; **RR-DU** avoids this through gradient alignment and localized noise at the unlearning user. Here $R := \mathrm{diam}(\Theta)$ and $R_{\mathrm{cert}} := \mathrm{diam}\big(\Theta \cap \mathbb{B}(\theta_{\mathrm{ref}}, \varrho)\big)$. Hidden constants and mild $\log T$ factors are absorbed in $\tilde{O}(\cdot)/\tilde{\Omega}(\cdot)$.

Method	Noise scale	Optimization term	Privacy term	Deletion capacity (scaling)
DDP (group privacy)	$\Theta\Big(\frac{mL}{\varepsilon}\sqrt{\frac{T\,\ln\left(\frac{1}{\delta}\right)\,\ln N}{N}}\Big)$	$\tilde{O}\Big(\frac{RL}{\sqrt{T}}\Big)$	$\tilde{O}\Big(R\,\frac{mL}{\varepsilon}\sqrt{\frac{d\,\ln\left(\frac{1}{\delta}\right)\,\ln N}{N}}\Big)$	$\tilde{\Omega}\Big(\frac{\varepsilon}{RL}\sqrt{\frac{N}{d\,\ln\left(\frac{1}{\delta}\right)\,\ln N}}\Big)$
RR-DU (ours)	$\Theta\Big(\frac{L}{\varepsilon}\sqrt{\frac{pT_u\,\ln\left(\frac{1}{\delta}\right)\,\ln N}{N}}\Big)$	$\tilde{O}\Big(\frac{R_{\mathrm{cert}}L}{\sqrt{sT_u}}\Big)$	$\tilde{O}\Big(R_{\mathrm{cert}}\,\frac{L}{\varepsilon}\,p\sqrt{\frac{d\,\ln\left(\frac{1}{\delta}\right)\,\ln N}{sN}}\Big)$	$\gamma > A:\ \Omega\Big(\frac{(\gamma-A)\,n_u}{L}\Big);\ \gamma \le A:\ 0$

Interpretation. Once the non-bias term A is pushed below γ (by increasing T_u, increasing s, choosing moderate p, and benefiting from larger N via the $\sqrt{\ln N/N}$ amplification), the capacity becomes $\Theta(\gamma n_u/L)$: it is linear in the local data size n_u and no longer improves with N. Any apparent N-gain seen in earlier formulas comes from the variance-limited regime where A dominates.

5.5. Comparison with DDP Baseline

Decentralized Differential Privacy (DDP) injects noise at every hop and relies on group privacy when certifying unlearning at edit distance m, which forces the DDP noise scale to grow linearly with m. In contrast, **RR-DU** concentrates noise at the unlearning user and uses gradient alignment so that the noise scale does not depend on m; the only m-dependence enters through the alignment bias term $O(L\,m/n_u)$. Table 2 summarizes the resulting scalings (convex case), separating optimization and privacy contributions to the excess-risk bound. The key takeaway is that, once the non-bias term A is below γ, **RR-DU** admits deletion capacity linear in n_u, while DDP is fundamentally limited by group privacy.

Key insight. DDP's deletion capacity is fundamentally limited by group privacy: its calibrated noise increases with m, so even under favorable mixing it cannot fully benefit from a large n_u. In contrast, **RR-DU** removes the direct m-dependence from the noise scale and incurs an m-related cost only through a controllable alignment bias. As a result, once A is reduced below γ, the achievable capacity scales as $\Theta(n_u)$. Under lightweight alignment, an additional approximation bias appears, and its effect grows with the deletion ratio m/n_u as well as with smoothness and heterogeneity, leading to a bias-dominated regime. This motivates a simple practical rule: use **lightweight** when m/n_u is small, and switch to **exact** alignment when m/n_u is larger and/or the data are strongly non-i.i.d.

When RR-DU dominates. **RR-DU** outperforms DDP precisely in the regime where the variance-driven term A is already below the target tolerance γ. Pushing A down can be accomplished by longer horizons T_u, modest local averaging s, and the natural network amplification $\sqrt{\ln N/N}$ (with $p \approx 1/N$), after which capacity becomes linear in n_u and independent of N. For more details on the deletion capacity of **RR-DU** and DDP, see Appendix C.4.

Practical notes. Increasing s reduces variance but does not alter privacy since noise is injected only at the unlearning user. Choosing $p \simeq 1/N$ aligns the random-walk visit rate with uniform mixing and keeps the corrective step well-weighted. For strongly convex and smooth nonconvex objectives, replace the optimization column in Table 2 by $\tilde{O}(L^2/(\mu s T_u))$ and $\tilde{O}(L^2/\sqrt{sT_u})$, respectively; the privacy and bias scalings remain unchanged.

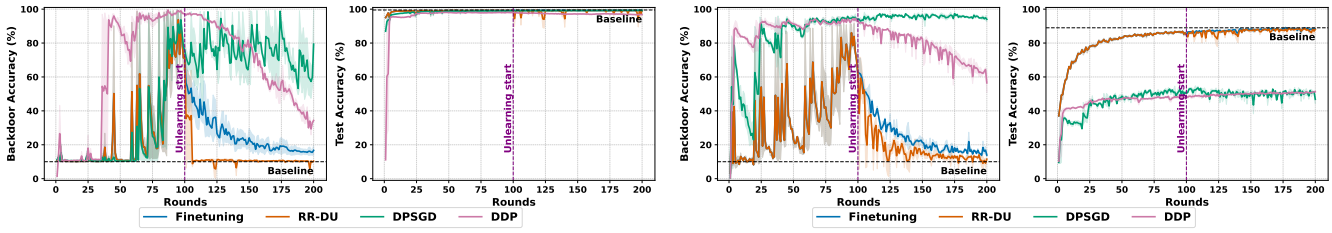
6. Experiments

6.1. Experimental Setup

Dataset and Models. We evaluate on two standard image classification benchmarks: CIFAR-10 [30] with ResNet-18 [25], and MNIST [31] with FLNet [36]. (See Appendix D for dataset/model details.)

Unlearning Scenario. Following prior work [2, 24], we use a backdoor (BadNets) setup [22] to assess unlearning effectiveness. We inject m=1000 poisoned samples into a *single* target client and train for T=100 token hops, then run unlearning for T_u=100 hops. We track backdoor accuracy and clean test accuracy; the desired trade-off is to drive backdoor accuracy to $\approx 10\%$ (random-guessing baseline) while maintaining high clean accuracy. To mirror real-world conditions, poisoning is performed in a decentralized manner, which can induce fluctuations of the backdoor metric during the initial training phase; our focus is on its reduction from the unlearning round onward.

Network and Hyperparameters. In the main experiments, we consider a complete graph with $N = 10$ clients and an i.i.d. uniform partition across clients. Unless stated otherwise, we set $\varepsilon = 1$ and $\delta = 10^{-5}$. Optimization is performed with Adam using step size $\eta = 0.005$, momentum $\lambda = 0.9$, and local averaging factor $s = 4$. The trust-region radius ϱ and effective gradient bound L are selected by grid search for each dataset-model pair, yielding $(\varrho, L) = (10.82, 0.5)$ for MNIST/FLNet and $(56.30, 0.2)$ for CIFAR-10/ResNet-18. Additional topology and scale robustness results, together with sensitivity analyses for ϱ,



(a) MNIST: backdoor and clean accuracy.

(b) CIFAR-10: backdoor and clean accuracy.

Figure 1. **Backdoor unlearning results on MNIST and CIFAR-10.** RR-DU vs. finetuning, DPSGD, and DDP. Vertical dashed lines mark unlearning start; horizontal dashed lines denote scratch baselines.

L, ε, and δ and an ablation of the routing probability p, are reported in Appendix E.

Baselines. For the main results, we compare against: (i) Decentralized DP (DDP) [14] under the same (ε, δ), with domain diameter R=10.0 and gradient bound L=1.0; (ii) DP-SGD [1] with clipping C=5.0; and (iii) fine-tuning after removing the poisoned data. We also consider PDUDT [42] in a limited matched setting. Since the two protocols differ fundamentally—PDUDT relies on dynamic topologies and gossip, whereas **RR-DU** follows a random walk on a fixed graph—we do not view the comparison as fully head-to-head. We therefore report this comparison separately, together with additional exact-mode, topology/scale, CIFAR-100, and efficiency results, in Appendix E.

6.2. Unlearning-Utility Trade-Off Evaluation

Figure 1 presents the evolution of backdoor accuracy and clean test accuracy on MNIST (**a**) and CIFAR-10 (**b**). The vertical purple dashed line marks the start of unlearning, and the black dashed horizontal line denotes the retraining-from-scratch baseline ($\approx 10\%$ backdoor accuracy for both datasets, $\approx 99.5\%$ clean accuracy on MNIST, and $\approx 89\%$ on CIFAR-10).

(a) MNIST. Before unlearning begins (round $\approx$ 100), all methods reach a high backdoor success rate ($> 90\%$), confirming that the attack is effective. After unlearning starts, **RR-DU** rapidly drives the backdoor accuracy down to the baseline level ($\approx 10\%$), while **finetuning** stabilizes slightly above it ($\approx 18\%$). **DPSGD** and **DDP** fail to forget the backdoor, plateauing around $\approx 60\%$ and $\approx 35\%$, respectively. On the clean set, RR-DU stays tightly concentrated around $\approx 99.1\% - 99.2\%$, matching finetuning and outperforming DDP, which drifts downward to $\approx 96.7\%$. Overall, RR-DU achieves the closest match to the scratch baseline in both utility and backdoor removal.

(b) CIFAR-10. A similar pattern emerges on CIFAR-10. Before unlearning, all methods fully learn the backdoor ($90\% - 100\%$ ASR). After unlearning starts, **RR-DU** again suppresses the backdoor aggressively, converging near the baseline ($\approx 10\%$). Finetuning remains above RR-DU ($\approx 25\% - 30\%$), while **DPSGD** and **DDP** retain even more backdoor signal. On clean accuracy, RR-DU climbs steadily toward the scratch baseline ($\approx 88\% - 89\%$), outperforming DPSGD and DDP, both of which saturate around $50\% - 55\%$ and never reach baseline performance. Finetuning recovers well but still exhibits worse backdoor removal than RR-DU.

Summary. Across both datasets, **RR-DU** consistently achieves the best trade-off: it removes the backdoor almost as effectively as retraining from scratch while maintaining near-optimal clean accuracy. In contrast, DP-based certifiers (DPSGD, DDP) retain significant backdoor signal and often sacrifice clean utility. Finetuning maintains clean accuracy but does not remove the backdoor nearly as well as **RR-DU**. Complementary results, including aligned exact-mode experiments, topology and scale robustness on CIFAR-10/ResNet18, a full-topology exact sweep, a CIFAR-100 extension, and a limited matched efficiency comparison with PDUDT, are provided in Appendix E.

7. Conclusion and Future Work

We introduced decentralized certified unlearning (DCU) through client *views* and proposed **RR-DU**, a random-walk mechanism that injects noise only at the deleting client while treating all other updates as post-processing. Under network differential privacy, we establish (ε, δ) guarantees on views, derive last-iterate and stationarity results for convex, strongly convex, and smooth nonconvex objectives, and characterize a two-regime deletion capacity governed by variance and alignment bias. Relative to DDP under the same privacy budget, **RR-DU** avoids group-privacy scaling with the forget-set size and reduces effective variance. Empirically, it approaches retraining-from-scratch backdoor removal while preserving strong clean accuracy. Current limitations include fixed graphs, a single token and deleting client, honest-but-curious observers, independent Gaussian noise, and predominantly i.i.d. data. Future work includes dynamic or gossip-based topologies, concurrent deletions, stronger heterogeneity, and communication-efficient designs.

References

[1] Martín Abadi, Andy Chu, Ian J. Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pages 308–318. ACM, 2016. 2, 8

[2] Manaar Alam, Hithem Lamri, and Michail Maniatakos. Get rid of your trail: Remotely erasing backdoors in federated learning. *IEEE Trans. Artif. Intell.*, 5(12):6683–6698, 2024. 1, 7

[3] Youssef Allouah, Anastasia Koloskova, Aymane El Firdoussi, Martin Jaggi, and Rachid Guerraoui. The privacy power of correlated noise in decentralized learning. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. 2

[4] Youssef Allouah, Joshua Kazdan, Rachid Guerraoui, and Sanmi Koyejo. The utility and complexity of in- and out-of-distribution machine unlearning. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. 1, 2, 3, 4, 18

[5] Borja Balle, Gilles Barthe, and Marco Gaboardi. Privacy amplification by subsampling: Tight analyses via couplings and divergences. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 6280–6290, 2018. 1, 2

[6] Borja Balle, James Bell, Adrià Gascón, and Kobbi Nissim. The privacy blanket of the shuffle model. In *CRYPTO*, 2019. 2

[7] Borja Balle, James Bell, Adrià Gascón, and Kobbi Nissim. Differentially private summation with multi-message shuffling, 2019. 2

[8] Aurélien Bellet, Edwige Cyffers, Davide Frey, Romaric Gaudel, Dimitri Lerévérend, and François Taïani. Unified privacy guarantees for decentralized learning via matrix factorization, 2025. 2

[9] Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Randomized gossip algorithms. *IEEE/ACM Transactions on Networking*, 14(SI):2508–2530, 2006. 2

[10] Yinzhi Cao and Junfeng Yang. Towards Making Systems Forget with Machine Unlearning. In *2015 IEEE Symposium on Security and Privacy, SP 2015, San Jose, CA, USA, May 17-21, 2015*, pages 463–480. IEEE Computer Society, 2015. 1

[11] Eli Chien, Haoyu Wang, Ziang Chen, and Pan Li. Langevin unlearning: A new perspective of noisy gradient descent for machine unlearning. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. 2

[12] Rishav Chourasia and Neil Shah. Forget unlearning: Towards true data-deletion in machine learning. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, pages 6028–6073. PMLR, 2023. 2

[13] Igor Colin, Aurélien Bellet, Joseph Salmon, and Stéphan Clémençon. Gossip dual averaging for decentralized optimization of pairwise functions. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pages 1388–1396. JMLR.org, 2016. 2

[14] Edwige Cyffers and Aurélien Bellet. Privacy amplification by decentralization. In *International Conference on Artificial Intelligence and Statistics, AISTATS 2022, 28-30 March 2022, Virtual Event*, pages 5334–5353. PMLR, 2022. 1, 2, 3, 4, 8, 14, 15, 17

[15] John C. Duchi, Michael I. Jordan, and Martin J. Wainwright. Local privacy and statistical minimax rates. In *FOCS*, 2013. 2

[16] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014. 1

[17] Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, and Kunal Talwar. Amplification by shuffling: From local to central differential privacy via anonymity. In *SODA*, 2019. 2

[18] European Commission. Data protection in the EU, 2024. https://commission.europa.eu/law/law-topic/data-protection/data-protection-eu_en. 1

[19] Vitaly Feldman, Ilya Mironov, Kunal Talwar, and Abhradeep Thakurta. Privacy amplification by iteration. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 521–532. IEEE Computer Society, 2018. 1, 2, 14, 15

[20] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting gradients – how easy is it to break privacy in federated learning? In *NeurIPS*, 2020. 2

[21] Antonio Ginart, Melody Y. Guan, Gregory Valiant, and James Zou. Making AI forget you: Data deletion in machine learning. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3513–3526, 2019. 2, 3

[22] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *CoRR*, abs/1708.06733, 2017. 7

[23] Chuan Guo, Tom Goldstein, Awni Y. Hannun, and Laurens van der Maaten. Certified data removal from machine learning models. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, pages 3832–3842. PMLR, 2020. 2, 3

[24] Anisa Halimi, Swanand Kadhe, Ambrish Rawat, and Nathalie Baracaldo. Federated unlearning: How to efficiently erase a client in fl? In *Updatable Machine Learning (part of ICML 2022), UpML 2022, Baltimore, USA, July 23, 2022*, 2022. 1, 7

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition,*

CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, pages 770–778. IEEE Computer Society, 2016. 7

[26] Yiyang Huang and Clément L. Canonne. Tight bounds for machine unlearning via differential privacy. *CoRR*, abs/2309.00886, 2023. 1, 4

[27] Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? In *FOCS*, 2008. 2

[28] Anastasia Koloskova, Nicolas Loizou, Sadra Boreiri, Martin Jaggi, and Sebastian U. Stich. A unified theory of decentralized sgd with changing topology and local updates. In *ICML*, 2020. 2

[29] Anastasia Koloskova, Youssef Allouah, Animesh Jha, Rachid Guerraoui, and Sanmi Koyejo. Certified unlearning for neural networks. *CoRR*, abs/2506.06985, 2025. 2

[30] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. 7

[31] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86(11):2278–2324, 1998. 7

[32] Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel sgd. In *NIPS*, 2017. 2

[33] Xiangru Lian, Wei Zhang, Ce Zhang, and Ji Liu. Asynchronous decentralized parallel stochastic gradient descent. In *ICML*, 2018. 2

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

[35] Ziyao Liu, Yu Jiang, Jiyuan Shen, Minyi Peng, Kwok-Yan Lam, Xingliang Yuan, and Xiaoning Liu. A survey on federated unlearning: Challenges, methods, and future directions. *ACM Comput. Surv.*, 57(1):2:1–2:38, 2025. 1

[36] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017. 1, 7

[37] Ilya Mironov. Rényi differential privacy. In *IEEE Computer Security Foundations Symposium (CSF)*, 2017. 2, 14, 15, 17, 18, 21

[38] Siqiao Mu and Diego Klabjan. Rewind-to-delete: Certified machine unlearning for nonconvex functions. *CoRR*, abs/2409.09778, 2024. 2

[39] Giovanni Neglia, Chuan Xu, Don Towsley, and Giacomo Calbi. Decentralized gradient methods: Does topology matter? In *AISTATS*, 2020. 2

[40] Office of the Attorney General, State of California. California Consumer Privacy Act (CCPA), 2024. https://oag.ca.gov/privacy/ccpa. 1

[41] Clément Pierquin, Aurélien Bellet, Marc Tommasi, and Matthieu Boussard. Privacy amplification through synthetic data: Insights from linear regression. *CoRR*, abs/2506.05101, 2025. 2

[42] Jing Qiao, Yu Liu, Zengzhe Chen, Mingyi Li, YUAN YUAN, Xiao Zhang, and Dongxiao Yu. PDUDT: Provable decentralized unlearning under dynamic topologies. In *Forty-second International Conference on Machine Learning*, 2025. 1, 2, 8, 26

[43] Ayush Sekhari, Jayadev Acharya, Gautam Kamath, and Ananda Theertha Suresh. Remember what you want to forget: Algorithms for machine unlearning. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 18075–18086, 2021. 1, 2, 3, 4, 18

[44] Ohad Shamir and Tong Zhang. Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. In *ICML*, 2013. 17, 20

[45] Reza Shokri, Marco Stronati, and Vitaly Shmatikov. Membership inference attacks against machine learning models. *CoRR*, abs/1610.05820, 2016. 2

[46] Youming Tao, Cheng-Long Wang, Miao Pan, Dongxiao Yu, Xiuzhen Cheng, and Di Wang. Communication efficient and provable federated unlearning. *Proc. VLDB Endow.*, 17(5): 1119–1131, 2024. 1

[47] Salil P. Vadhan. The complexity of differential privacy. In *Tutorials on the Foundations of Cryptography*, pages 347–450. Springer International Publishing, 2017. 2

[48] Martin Van Waerebeke, Marco Lorenzi, Giovanni Neglia, and Kevin Scaman. When to forget? complexity trade-offs in machine unlearning. *CoRR*, abs/2502.17323, 2025. 1, 2

[49] Leijie Wu, Song Guo, Junxiao Wang, Zicong Hong, Jie Zhang, and Yaohong Ding. Federated unlearning: Guarantee the right of clients to forget. *IEEE Netw.*, 36(5):129–135, 2022. 1

[50] Guanhua Ye, Tong Chen, Quoc Viet Hung Nguyen, and Hongzhi Yin. Heterogeneous decentralised machine unlearning with seed model distillation. *CAAI Trans. Intell. Technol.*, 9(3):608–619, 2024.

[51] Yanli Yuan, Bingbing Wang, Chuan Zhang, Zehui Xiong, Chunhai Li, and Liehuang Zhu. Toward efficient and robust federated unlearning in iot networks. *IEEE Internet of Things Journal*, 11(12):22081–22090, 2024. 1

[52] Binchi Zhang, Yushun Dong, Tianhao Wang, and Jundong Li. Towards certified unlearning for deep neural networks. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. 2

[53] Lefeng Zhang, Tianqing Zhu, Haibin Zhang, Ping Xiong, and Wanlei Zhou. Fedrecovery: Differentially private machine unlearning for federated learning frameworks. *IEEE Trans. Inf. Forensics Secur.*, 18:4732–4746, 2023. 1