Fully Decentralized Certified Unlearning

Supplementary Material

A. Additional Notation, Definitions, and Rényi-DP Tools

This section summarizes the notation used in the main paper, restates the unlearning and Network-DP definitions, and collects the Rényi-DP tools used in the analysis of the DDP (NetDP) baseline and RR-DU.

Symbol	Description
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	Communication graph.
$\mathcal{V} = \{1, \dots, N\}$	Set of users (clients).
$\mathcal{E}$	Edge set of the graph.
$N = \lvert\mathcal{V}\rvert$	Number of users.
$u, v \in \mathcal{V}$	User indices.
$(u, v) \in \mathcal{E}$	Edge between users u and v.
$\mathcal{N}_u$	The set of neighbour of the node u.
$D_u \subseteq \mathcal{Z}$	Local dataset of user u.
$n_u = \lvert D_u\rvert$	Local dataset size at user u.
$D = \bigcup_{u\in\mathcal{V}} D_u$	Global dataset across all users.
$n = \lvert D\rvert = \sum_{u\in\mathcal{V}} n_u$	Global dataset size.
$D_f \subseteq D_u$	Forget (delete) set at user u.
$m = \lvert D_f\rvert$	Size of the forget set.
$D_{u\setminus f} = D_u \setminus D_f$	Remaining data at user u after deletion.
$n_{u\setminus f} = \lvert D_{u\setminus f}\rvert$	Size of $D_{u\setminus f}$.
$\mathcal{Z}$	Data space.
$\mathcal{P}$	Data distribution over $\mathcal{Z}$.
$D \sim \mathcal{P}^n$	Dataset drawn i.i.d. from $\mathcal{P}$.

Table 1. Graph, users, and datasets.

Unlearning and Network-DP Definitions

Definition A.1 ((ε, δ)-Certified Unlearning (global))**.** 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 $\lvert D_f\rvert \le m$. 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 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 $\theta \subseteq \Theta$:

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

Definition A.2 (Network Differential Privacy (Network-DP))**.** An algorithm $\mathcal{A}$ satisfies (ε, δ)-network DP if for all distinct $u, v \in \mathcal{V}$, all u-neighboring datasets $D \sim_u D'$, and all measurable sets $\theta \subseteq \Theta_v$,

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

Definition A.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 $\theta \subseteq \Theta_v$,

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

and the same inequality holds with $\mathcal{U}$ and $\mathcal{C}$ swapped.

Symbol	Description
$\theta \in \Theta \subseteq \mathbb{R}^d$	Model parameter vector.
d	Model dimension (number of parameters).
Θ	Feasible parameter domain.
θ_0	Initial model before training / unlearning.
θ_t	Model after t updates.
θ_T	Model after T updates (final iterate).
θ^*	Population risk minimizer.
θ_{ref}	Trust-region center.
ϱ	Trust-region radius around θ_{ref}.
$B(\theta_{\mathrm{ref}}, \varrho)$	Ball $\{\theta : \|\theta - \theta_{\mathrm{ref}}\|_2 \leq \varrho\}$.
Θ_{cert}	Certification domain $\Theta \cap B(\theta_{\mathrm{ref}}, \varrho)$.
$\ell(\theta; z)$	Per-example loss at data point z.
$\ell_u(\theta)$	Local objective at user u on D_u.
$\ell_{u\setminus f}(\theta)$	Local objective at user u on $D_{u\setminus f}$.
$\ell_f(\theta)$	Average loss over the forget set D_f.
$\mathcal{L}(\theta)$	Population risk $\mathbb{E}_{z\sim\mathcal{P}}[\ell(\theta; z)]$.
$\mathcal{L}^*$	Optimal population risk value.
$\mathcal{L}(\theta, \mathcal{V})$	Empirical risk on the global dataset D.

Table 2. Model parameters, losses, and risks.

Symbol	Description
$\mathcal{A}$	Training algorithm (e.g., token Net-SGD or DDP NetDP).
$\mathcal{U}$	Unlearning algorithm (e.g., RR-DU).
$\mathcal{C}$	Certifying algorithm in unlearning definitions.
$T(D)$	Auxiliary statistics of D used by $\mathcal{U}$ or $\mathcal{C}$.
$\mathcal{A}(D)$	Transcript or output of $\mathcal{A}$ on D.
$O_u(\mathcal{A}(D))$	View (partial transcript) observed by user u.
$\Theta_u = \mathrm{Range}(O_u)$	Observation space of user u's views.
$D \sim_u D'$	Datasets differing only in user u's data.
ε	Privacy / unlearning parameter (multiplicative).
δ	Privacy / unlearning parameter (additive slack).
γ	Excess-risk tolerance in deletion capacity.
$m_{\varepsilon,\delta}^{\mathcal{A},\mathcal{U}}(d, N)$	Deletion capacity of $(\mathcal{A}, \mathcal{U})$ at (ε, δ).

Table 3. Algorithms, views, and privacy-related notation.

Definition A.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 (in either the global or decentralized sense above), and for a fixed tolerance $\gamma > 0$, 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 (recap). Differential Privacy implies global certified unlearning with certifier $\mathcal{C}(D \setminus D_f) = \mathcal{A}(D \setminus D_f)$, and Network-DP implies decentralized certified unlearning on views with the same certifier. These reductions underpin the deletion-capacity guarantees for the DDP NetDP baseline.

Symbol	Description
R	Diameter of Θ: $R := \sup_{\theta,\theta' \in \Theta} \|\theta - \theta'\|_2$.
R_{cert}	Diameter of $\Theta_{\mathrm{cert}} := \Theta \cap B(\theta_{\mathrm{ref}}, \varrho)$, i.e., $R_{\mathrm{cert}} := \sup_{\theta,\theta' \in \Theta_{\mathrm{cert}}} \|\theta - \theta'\|_2 \le 2\varrho$.
L	Lipschitz / smoothness constant of the loss (and clipping threshold).
μ	Strong-convexity constant (when assumed).
G^2	Upper bound on $\mathbb{E}[\|g_t\|_2^2]$ (gradient second moment).
σ^2	Gaussian noise variance per coordinate.
σ	Gaussian noise standard deviation.
α	Rényi-DP order in privacy analysis.
κ	Bound on inverse Hessian / condition number (when used).
η	Constant stepsize in SGD / token updates.
η_t	Stepsize at iteration / hop t.
T	Number of training token hops / rounds.
T_u	Number of unlearning token hops / rounds.
t	Iteration / hop index.
s	Local averaging factor (minibatches per token visit).
p	Routing probability toward the unlearning user.
g_t	Stochastic gradient estimate at step t.
Z_t	Gaussian noise vector added at step t.
B_u	Minibatch sampled from D_u.
B_f	Minibatch sampled from the forget set D_f.

Table 4. Optimization and analysis constants.

Symbol / Name	Description
RR-DU	Randomized-Restart Decentralized Unlearning algorithm.
DDP NetDP	Decentralized-DP baseline (network-private SGD).
DP-SGD	Differentially-private SGD baseline.
Fine-tuning	Retraining baseline from the pre-unlearning model (no noise).
FLNet	Lightweight convolutional network used on MNIST.
ResNet-18	Residual network used on CIFAR-10.
clean acc.	Test accuracy on clean (unpoisoned) examples.
ASR (backdoor acc.)	Test accuracy on backdoor-triggered examples.
y^{bd}	Target label for backdoor (poisoned) samples.

Table 5. Baselines, models, and evaluation metrics.

A.1. Rényi-DP and Network-RDP Tools

We now collect the Rényi-DP tools used in the DDP NetDP and RR-DU analysis, following Mironov [**?**], Feldman et al. [**?**], and the NetDP token-SGD analysis in [**?**].

Definition A.5 (Rényi divergence [**?**])**.** Let $1 < \alpha < \infty$ and let μ, ν be probability measures such that μ is absolutely continuous with respect to ν. The Rényi divergence of order α between μ and ν is

$$D_\alpha(\mu\|\nu) \;:=\; \frac{1}{\alpha - 1} \log \int \Big(\frac{d\mu}{d\nu}(z)\Big)^{\alpha} d\nu(z).$$

If $U \sim \mu$ and $V \sim \nu$, we often write $D_\alpha(U\|V)$ for $D_\alpha(\mu\|\nu)$.

Definition A.6 (Rényi Differential Privacy [**?**])**.** A randomized mechanism $\mathcal{M}$ with domain $\mathcal{X}^n$ and range $\mathcal{Y}$ is said to satisfy (α, ε)-Rényi Differential Privacy if for all neighboring datasets $D, D' \in \mathcal{X}^n$ and all $1 < \alpha < \infty$,

$$D_\alpha\big(\mathcal{M}(D) \,\big\|\, \mathcal{M}(D')\big) \;\le\; \varepsilon.$$

Definition A.7 (Network Rényi-DP)**.** A decentralized algorithm $\mathcal{A}$ on a graph with views $O_v(\mathcal{A}(D))$ satisfies (α, ε)-*network Rényi-DP* if, for every pair of distinct users $u, v \in \mathcal{V}$ and every u-neighboring datasets $D \sim_u D'$,

$$D_\alpha\big(O_v(\mathcal{A}(D)) \,\big\|\, O_v(\mathcal{A}(D'))\big) \;\le\; \varepsilon.$$

Proposition A.8 (RDP $\Rightarrow$ (ε, δ)-DP conversion [**?**])**.** *If a mechanism $\mathcal{M}$ satisfies (α, ε)-RDP for some $\alpha > 1$, then for every $\delta \in (0, 1)$ it satisfies (ε', δ)-DP with*

$$\varepsilon' \;=\; \varepsilon + \frac{\log(1/\delta)}{\alpha - 1}.$$

The same implication holds in the network setting by applying this bound to each pair of neighboring datasets and each view.

Theorem A.9 (RDP of PNSGD [**? ?**])**.** *Let $W \subset \mathbb{R}^d$ be convex and let $\{f(\cdot; x)\}_{x \in \mathcal{X}}$ be a family of convex, L-Lipschitz and β-smooth functions on W. Consider Projected Noisy Stochastic Gradient Descent (PNSGD) on dataset $D = (x_1, \dots, x_n)$:*

$$w_{t+1} = \Pi_W\big(w_t - \eta(\nabla f(w_t; x_{t+1}) + Z)\big), \qquad Z \sim \mathcal{N}(0, \sigma^2 I_d),$$

run for n steps with stepsize $\eta \le 2/\beta$. Then, for any order $\alpha > 1$ and any $t \in \{1, \dots, n\}$, the mechanism satisfies (α, ε_t)-RDP with respect to the t-th input, where

$$\varepsilon_t \;=\; \frac{\alpha \cdot 2L^2}{\sigma^2(n + 1 - t)}.$$

Proposition A.10 (Weak convexity of Rényi divergence [**?** , Appendix A])**.** *Let $\mu_1, \dots, \mu_m$ and $\nu_1, \dots, \nu_m$ be probability distributions on a common measurable space, and suppose that for some $c \in (0, 1]$ and all $i \in [m]$,*

$$D_\alpha(\mu_i \| \nu_i) \;\le\; \frac{c}{\alpha - 1}.$$

Let ρ be a distribution on $[m]$ and define μ_ρ (resp. ν_ρ) as the mixture distribution obtained by first sampling $i \sim \rho$ and then a sample from μ_i (resp. ν_i). Then

$$D_\alpha(\mu_\rho \| \nu_\rho) \;\le\; (1 + c) \cdot \mathbb{E}_{i \sim \rho}\big[D_\alpha(\mu_i \| \nu_i)\big].$$

Lemma A.11 (View-level RDP for token SGD on complete graphs [**?** , Theorem 4])**.** *Consider token-based SGD with Gaussian noise $\mathcal{N}(0, \sigma^2 I_d)$ on a complete graph with N users, as in the DDP NetDP analysis of [**?**]. Let T_u be the number of visits to user u, and fix distinct users $u \ne v$. Then, for each order $\alpha > 1$, there exists an absolute constant $C > 0$ such that for any pair of u-neighboring datasets $D \sim_u D'$,*

$$D_\alpha\big(Y_v \| Y_v'\big) \;\le\; C \cdot \frac{\alpha L^2\, T_u\, \ln N}{\sigma^2\, N},$$

where Y_v and Y_v' denote the random views of user v under D and D', respectively.

Remark A.12. Lemma A.11 is the network version of amplification by iteration for PNSGD (Theorem A.9), combined with the random walk structure and weak convexity of Rényi divergence (Proposition A.10) as in [**?**].

Proposition A.13 (Network-RDP $\Rightarrow$ Network-DP)**.** *If a decentralized algorithm $\mathcal{A}$ satisfies (α, ε)-network RDP in the sense of Definition A.7, then for every $\delta \in (0, 1)$ it satisfies (ε', δ)-network DP with*

$$\varepsilon' \;=\; \varepsilon + \frac{\log(1/\delta)}{\alpha - 1}.$$

Proof. Apply Proposition A.8 to each pair of neighboring datasets $D \sim_u D'$ and each view $O_v(\mathcal{A}(D))$ separately. □

B. Algorithms and Implementation Details

This section provides the pseudocode for the two auxiliary procedures referenced in the main paper: (i) the token-based Network-SGD training phase, which produces the pre-unlearning model, and (ii) the Network-Private SGD baseline implementing DDP NetDP. Both algorithms are stated in the notation of Section A.

B.1. Network-SGD (Token Training)

Network-SGD is the basic decentralized training routine used to obtain the initial model θ_0 before any deletion request. A single token carries the current parameter vector θ and performs a random walk over the user graph. Each time the token visits a user u, that user computes a local stochastic gradient step on its dataset D_u and then forwards the token to a uniformly random neighbor.

Algorithm 1 makes this process explicit: it takes as input the communication graph $\mathcal{G}$, an initial token location u_0, the local datasets $\{D_u\}_{u\in\mathcal{V}}$, and runs for T token hops. The final model θ returned by Algorithm 1 is used as the starting point for both the DDP NetDP baseline and RR-DU in the experiments.

Algorithm 1 Network-SGD (token-based decentralized training)

Require: Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, initial token location $u_0 \in \mathcal{V}$, stepsize $\eta > 0$, horizon $T \in \mathbb{N}$, local datasets $\{D_u\}_{u\in\mathcal{V}}$

Initialize model $\theta \in \Theta \subseteq \mathbb{R}^d$
$u \leftarrow u_0$
for $t = 1$ to T **do**
  Sample minibatch $B_u \subseteq D_u$
  $g_t \leftarrow \nabla_\theta \ell(\theta; B_u)$
  $\theta \leftarrow \theta - \eta\, g_t$
  Sample neighbor $v \sim \mathrm{Unif}\{w \in \mathcal{V} : (u, w) \in \mathcal{E}\}$
  $u \leftarrow v$ ▷ forward the token
end for
return θ

B.2. Network-Private SGD (DDP NetDP Baseline)

Network-Private SGD is the decentralized-DP baseline (DDP NetDP) used in our deletion-capacity analysis. It has the same token structure as Network-SGD but adds Gaussian noise to each update and projects back onto the feasible set Θ. The noise scale is chosen as a function of (ε, δ) and the Lipschitz bound L, following the DDP calibration used in the main paper.

Algorithm 2 summarizes this procedure. At each token hop, a user u is sampled uniformly from $\mathcal{V}$, a minibatch B_u is drawn, and a noisy gradient $g_t + Z_t$ is applied before projecting onto Θ. This is the algorithm used as the DDP NetDP baseline in the experiments and as the certifier in the theoretical results on decentralized DP deletion capacity.

Algorithm 2 Network-Private SGD (DDP NetDP baseline)

Require: Convex set $\Theta \subset \mathbb{R}^d$, stepsize $\eta > 0$, horizon $T \in \mathbb{N}$, gradient bound L (i.e., $\|\nabla_\theta \ell(\theta; z)\|_2 \le L$ for all z), target privacy (ε, δ), datasets $\{D_u\}_{u\in\mathcal{V}}$

Initialize $\theta \in \Theta$
$\sigma^2 \leftarrow \dfrac{8L^2 \ln\!\big(1.25/\delta\big)}{\varepsilon^2}$
for $t = 1$ to T **do**
  Draw $u \sim \mathrm{Unif}(\mathcal{V})$ ▷ token visit
  Sample minibatch $B_u \subseteq D_u$
  $g_t \leftarrow \nabla_\theta \ell(\theta; B_u)$
  Draw $Z_t \sim \mathcal{N}(0, \sigma^2 I_d)$
  $\theta \leftarrow \Pi_\Theta\big(\theta - \eta\,(g_t + Z_t)\big)$
end for
return θ

C. Proofs of Main Results

This section (Section C in the supplementary material) contains detailed proofs of the main theoretical results. We first give the full proof of the deletion-capacity bound for the DDP NetDP baseline (Theorem 3.5), then discuss the effect of decentralization and local averaging, and finally collect our optimization assumptions and RR-DU utility / capacity analysis.

C.1. Proof of Theorem 3.5 (Deletion Capacity of DDP NetDP)

We start by recalling the reduction from Network-DP (Definition A.2) to decentralized certified unlearning (Definition A.3), together with a standard group-privacy bound. These are used in the deletion-capacity proof for the DDP NetDP baseline [**?**].

Network-DP implies DCU and group privacy. Let $\mathcal{A}$ be a decentralized algorithm on a graph that satisfies $(\varepsilon_0, \delta_0)$-network DP in the sense of Definition A.2. Recall that $O_v(\mathcal{A}(D))$ denotes the view of user v and $\Theta_v := \mathrm{Range}(O_v)$.

Proposition C.1 (Network-DP implies decentralized certified unlearning)**.** *Let $\mathcal{A}$ satisfy $(\varepsilon_0, \delta_0)$-network DP. Define $\mathcal{U}(D_f, \mathcal{A}(D)) := \mathcal{A}(D)$ and $\mathcal{C}(D \setminus D_f) := \mathcal{A}(D \setminus D_f)$. Assume D and $D \setminus D_f$ differ only in the data of a single user u. Then, for any $v \neq u$, the pair $(\mathcal{A}, \mathcal{U})$ achieves $(\varepsilon_0, \delta_0)$ decentralized certified unlearning on views in the sense of Definition A.3.*

Proof. Fix $u \in \mathcal{V}$ and $D_f \subseteq D_u$, and set $D' := D \setminus D_f$. Then $D \sim_u D'$. For any $v \neq u$ and measurable $\theta \subseteq \Theta_v$, network DP gives

$$\mathbb{P}\big[\, O_v(\mathcal{A}(D)) \in \theta \,\big] \leq e^{\varepsilon_0}\, \mathbb{P}\big[\, O_v(\mathcal{A}(D')) \in \theta \,\big] + \delta_0.$$

Using $O_v(\mathcal{U}(D_f, \mathcal{A}(D))) = O_v(\mathcal{A}(D))$ and $O_v(\mathcal{C}(D \setminus D_f)) = O_v(\mathcal{A}(D'))$ yields the first inequality in Definition A.3. Swapping D and D' gives the reverse inequality. □

For the dependence on the edit distance m (size of the forget set), we use the usual group-privacy bound, obtained via Rényi DP and advanced composition (see, e.g., [**?**]).

Lemma C.2 (Group privacy)**.** *Suppose a mechanism (e.g., DDP NetDP) satisfies $(\varepsilon_0, \delta_0)$-network DP with respect to a change in a single user's data. Then for any integer $m \geq 1$, it satisfies $(\varepsilon_m, \delta_m)$-network DP for changes in up to m users' data with*

$$\varepsilon_m \;\leq\; \sqrt{2m \log(1/\tilde{\delta})}\, \varepsilon_0, \qquad \delta_m \;\leq\; m\delta_0 + \tilde{\delta}, \tag{1}$$

for any choice of $\tilde{\delta} > 0$.

In the deletion-capacity setting of Definition A.4, the edit distance m corresponds to removing up to m points in the forget set D_f; Lemma C.2 quantifies how the privacy parameters degrade as m grows.

Proof of Theorem 3.5. We now derive the deletion-capacity lower bound for the DDP NetDP baseline, following the sketch in the main text.

Step 1: Utility of projected noisy token-SGD. On a bounded domain, the projected noisy-SGD bound of Lemma C.5 (with $\Theta_{\mathrm{dom}} = \Theta$ and $R_{\mathrm{dom}} = R$) implies that, for a suitable iterate θ_T produced by projected noisy token-SGD,

$$\mathbb{E}\big[\mathcal{L}(\theta_T) - \mathcal{L}(\theta^*)\big] \;\leq\; \frac{R^2}{2\eta T} \;+\; \frac{\eta}{2}\big(G^2 + d\sigma^2\big),$$

and, using Corollary C.6, there exists a choice of stepsizes such that

$$\mathbb{E}\big[\mathcal{L}(\theta_T) - \mathcal{L}(\theta^*)\big] \;\leq\; \frac{2R\,G\,(2 + \log T)}{\sqrt{T}},$$

with effective variance $G^2 = L^2 + d\sigma^2$, where L bounds the gradient norm and σ^2 is the Gaussian noise variance. These results are standard in projected (noisy) SGD; see, e.g., [**?**].

Step 2: Decentralized view calibration via network-RDP. Consider DDP NetDP (network-private token-SGD) on a complete graph. Let Y_v denote the view of user v. The view-based Rényi-DP analysis of token-SGD in [**?**], combined with the network-RDP definition (Definition A.7) and tools from Section A.1, implies that the algorithm satisfies $(\alpha, \varepsilon_\alpha)$-network RDP with

$$\varepsilon_\alpha \;\lesssim\; \frac{\alpha L^2 T_u \ln N}{\sigma^2 N}, \qquad T_u \approx \frac{T}{N},$$

where T_u is the expected number of contributions per user and $N = |\mathcal{V}|$. Converting this network-RDP bound to $(\varepsilon_0, \delta_0)$-network DP by Proposition A.13 (which itself relies on Proposition A.8) and optimizing over the order α as in [**?**] yields, up to constants,

$$\varepsilon_0 \;\approx\; \frac{L}{\sigma}\sqrt{\frac{T\,\ln N}{N\,\ln(1/\delta_0)}}.$$

Step 3: Group privacy for m deletions. To target (ε, δ) at edit distance m, we apply Lemma C.2 to the view-level guarantee. In the worst case, the forget set induces an m-fold change, so the base algorithm must satisfy approximately $(\varepsilon_m, \delta_m) \approx (\varepsilon, \delta)$, which we implement via the simple choice

$$\varepsilon_0 = \varepsilon/m, \qquad \delta_0 = \delta/m.$$

Substituting these into the calibration above yields a required noise scale

$$\sigma_{\text{DDP}} \;\approx\; \frac{mL}{\varepsilon}\sqrt{\frac{T\,\ln N\,\ln(1/\delta)}{N}}.$$

Step 4: Utility in the privacy-dominated regime. In the regime where the privacy noise dominates, $d\sigma^2 \gg L^2$, we have $G \approx \sqrt{d}\,\sigma_{\text{DDP}}$. Plugging this into the utility bound from Step 1 gives

$$\mathbb{E}\big[\mathcal{L}(\theta_T) - \mathcal{L}(\theta^*)\big] \;\lesssim\; \frac{2R\sqrt{d}\,\sigma_{\text{DDP}}\,(2+\log T)}{\sqrt{T}}.$$

Substituting the expression for σ_{DDP},

$$\mathbb{E}\big[\mathcal{L}(\theta_T) - \mathcal{L}(\theta^*)\big] \;\lesssim\; \frac{2R\,L\,m}{\varepsilon}\sqrt{\frac{d\,\ln(1/\delta)\,\ln N}{N}}\,(2+\log T).$$

Step 5: Solving for the deletion capacity m. By Definition A.4, the deletion capacity is the largest m such that the expected excess risk remains below γ:

$$\mathbb{E}\big[\mathcal{L}(\theta_T) - \mathcal{L}(\theta^*)\big] \;\le\; \gamma.$$

Imposing this inequality and rearranging for m gives

$$m \;\gtrsim\; \frac{\varepsilon}{R\,L\,(2+\log T)}\sqrt{\frac{N}{d\,\ln(1/\delta)\,\ln N}},$$

up to absolute constants and polylogarithmic factors, which yields the $\widetilde{\Omega}(\cdot)$ scaling stated as Theorem 3.5 in the main paper. □

C.2. Decentralization Effect on Deletion Capacity and Local Averaging

We now contrast the decentralized deletion-capacity bound of Theorem 3.5 with its centralized (curator-DP) analogue, and highlight the effect of local averaging on the decentralized bound. The comparison closely parallels the analysis of centralized DP-SGD in [**? ?**].

Centralized DP-SGD. In curator (centralized) DP-SGD on a dataset of size n, standard analyses on bounded domains give excess risk scaling as $O(RL/\sqrt{n})$ once the Gaussian noise is calibrated to the target (ε, δ) (see, e.g., [**?**]). Combining this with the deletion-capacity criterion of Definition A.4 yields

$$m^{\text{central}}_{\varepsilon,\delta}(d, n) \;=\; \widetilde{\Omega}\left(\frac{\varepsilon\, n}{R\,L\,\sqrt{d\,\log(1/\delta)}}\right),$$

up to logarithmic factors.

Decentralized DDP NetDP baseline. By contrast, the decentralized DDP NetDP baseline depends on the number of clients N and only logarithmically on the number of token hops T:

$$m_{\varepsilon,\delta}^{\text{decentral}}(d,N) \;=\; \widetilde{\Omega}\left(\frac{\varepsilon}{R\,L\,(2+\log T)}\sqrt{\frac{N}{d\,\log(1/\delta)\,\log N}}\right),$$

as derived in Section C.1. Operationally, T tracks the number of effective stochastic updates seen across the network: with minibatch size b and κ passes over users' data, a typical scaling is $T \approx \kappa \sum_u n_u/b$, and each user contributes $T_u \approx T/N$ updates in expectation. The factors $\sqrt{N}$ and $\sqrt{\ln N}$ reflect the network-DP amplification on the complete graph established by the view-RDP bound in Lemma A.11.

Effect of local averaging. If at each token hop the outgoing message averages $s \geq 1$ independent gradients with independent Gaussian noise *before* any observer's first view, then the effective variance in the utility bound becomes

$$G^2 \;=\; L^2 \;+\; \frac{d\,\sigma^2}{s}.$$

With the same privacy calibration for σ, the capacity expression gains a factor $\sqrt{s}$:

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

provided the view-DP accounting treats each averaged message as a single observation epoch. Two special cases:

- *Token passing with one update per hop* ($s = 1$) recovers the bound in Theorem 3.5.
- *Synchronous rounds that average N users* before any observation correspond to $s \approx N$, yielding an extra $\sqrt{N}$ factor in utility (and thus capacity) relative to one-update-per-hop token passing.

C.3. Optimization Assumptions and Preliminary Lemmas

We next collect the optimization assumptions and utility lemmas used both for the DDP NetDP baseline and for RR-DU. Throughout, we work with the population risk $\mathcal{L}(\theta) = \mathbb{E}_{z\sim\mathcal{P}}[\ell(\theta; z)]$.

Assumption C.3 (Convexity and smoothness)**.** Let $\Theta_{\text{dom}} \subset \mathbb{R}^d$ be convex with diameter $R_{\text{dom}} := \sup_{\theta,\theta'\in\Theta_{\text{dom}}} \|\theta-\theta'\|_2 < \infty$, and let $\mathcal{L} : \Theta_{\text{dom}} \to \mathbb{R}$ be convex and L-smooth, i.e., for all $\theta, \theta' \in \Theta_{\text{dom}}$,

$$\mathcal{L}(\theta') \leq \mathcal{L}(\theta) + \langle \nabla\mathcal{L}(\theta), \theta' - \theta\rangle + \frac{L}{2}\|\theta' - \theta\|_2^2.$$

Assumption C.4 (Strong convexity)**.** In the strongly convex case, we additionally assume that $\mathcal{L}$ is μ-strongly convex on Θ_{dom}, i.e., for all $\theta, \theta' \in \Theta_{\text{dom}}$,

$$\mathcal{L}(\theta') \geq \mathcal{L}(\theta) + \langle \nabla\mathcal{L}(\theta), \theta' - \theta\rangle + \frac{\mu}{2}\|\theta' - \theta\|_2^2.$$

We first recall a utility bound for projected noisy SGD on a bounded domain, stated directly in terms of $\mathcal{L}$.

Lemma C.5 (Utility of projected noisy SGD)**.** *Let Assumption C.3 hold. Consider*

$$\theta_{t+1} \;=\; \Pi_{\Theta_{\text{dom}}}\big(\theta_t - \eta(g_t + Z_t)\big),$$

where g_t is an unbiased estimator of $\nabla\mathcal{L}(\theta_t)$ with $\mathbb{E}[\|g_t\|_2^2] \leq G^2$, and $Z_t \sim \mathcal{N}(0, \sigma^2 I_d)$ i.i.d. Let $\eta_t \equiv \eta \leq 1/L$ and define $\bar\theta_T := \frac{1}{T}\sum_{t=1}^T \theta_t$. Then, for any minimizer θ^* of $\mathcal{L}$ in Θ_{dom},

$$\mathbb{E}\big[\mathcal{L}(\bar\theta_T) - \mathcal{L}(\theta^*)\big] \;\leq\; \frac{R_{\text{dom}}^2}{2\eta T} \;+\; \frac{\eta}{2}\big(G^2 + d\sigma^2\big). \tag{2}$$

Proof. The proof is standard (see, e.g., [**?**]), combining L-smoothness, non-expansiveness of the projection, and the unbiasedness of g_t. For completeness, we sketch the argument.

By L-smoothness of $\mathcal{L}$, for any t,

$$\mathcal{L}(\theta_{t+1}) \le \mathcal{L}(\theta_t) + \langle \nabla\mathcal{L}(\theta_t), \theta_{t+1} - \theta_t \rangle + \frac{L}{2}\|\theta_{t+1} - \theta_t\|_2^2.$$

Using the update $\theta_{t+1} = \Pi_{\Theta_{\mathrm{dom}}}(\theta_t - \eta(g_t + Z_t))$ and the non-expansiveness of $\Pi_{\Theta_{\mathrm{dom}}}$, one obtains

$$\langle \nabla\mathcal{L}(\theta_t), \theta_{t+1} - \theta_t \rangle \le -\eta\langle \nabla\mathcal{L}(\theta_t), g_t + Z_t \rangle + \frac{\eta^2}{2}\|\nabla\mathcal{L}(\theta_t)\|_2^2.$$

Taking expectation conditional on θ_t and using $\mathbb{E}[g_t \mid \theta_t] = \nabla\mathcal{L}(\theta_t)$ and $\mathbb{E}[Z_t] = 0$, we find

$$\mathbb{E}\big[\mathcal{L}(\theta_{t+1}) \mid \theta_t\big] \le \mathcal{L}(\theta_t) - \frac{\eta}{2}\|\nabla\mathcal{L}(\theta_t)\|_2^2 + \frac{\eta^2 L}{2}\,\mathbb{E}\big[\|g_t + Z_t\|_2^2 \mid \theta_t\big].$$

Using $\mathbb{E}[\|g_t + Z_t\|_2^2] \le 2G^2 + 2d\sigma^2$ and $\mathcal{L}(\theta_t) - \mathcal{L}(\theta^*) \le \frac{1}{2L}\|\nabla\mathcal{L}(\theta_t)\|_2^2$ and summing over $t = 1, \dots, T$ yields

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\big[\mathcal{L}(\theta_t) - \mathcal{L}(\theta^*)\big] \le \frac{R_{\mathrm{dom}}^2}{2\eta T} + \frac{\eta}{2}(G^2 + d\sigma^2),$$

since $\|\theta_1 - \theta^*\|_2 \le R_{\mathrm{dom}}$ and $\|\theta_{T+1} - \theta^*\|_2^2 \ge 0$. Convexity of $\mathcal{L}$ then gives $\mathcal{L}(\bar{\theta}_T) \le \frac{1}{T}\sum_t \mathcal{L}(\theta_t)$ and taking expectations yields (2). □

Corollary C.6 (Bound with $(2 + \log T)/\sqrt{T}$)**.** *Under the assumptions of Lemma C.5, there exists a non-increasing stepsize schedule $\{\eta_t\}_{t=1}^T$ with $\eta_t \le 1/L$ such that, for a suitable iterate θ_T,*

$$\mathbb{E}\big[\mathcal{L}(\theta_T) - \mathcal{L}(\theta^*)\big] \;\le\; \frac{2R_{\mathrm{dom}}G(2 + \log T)}{\sqrt{T}},$$

with $G^2 = L^2 + d\sigma^2$.

Proof. This follows from standard analyses of projected SGD with decreasing stepsize, e.g., $\eta_t = \min\{1/L, R_{\mathrm{dom}}/(G\sqrt{t})\}$, together with a doubling-trick argument to select an iterate with the desired bound; see, for instance, [**?**]. □

In the sequel we apply Lemma C.5 and Corollary C.6 in two regimes:

- For the DDP NetDP baseline, $\Theta_{\mathrm{dom}} = \Theta$ and $R_{\mathrm{dom}} = R$.
- For RR-DU, $\Theta_{\mathrm{dom}} = \Theta_{\mathrm{cert}}$ and $R_{\mathrm{dom}} = R_{\mathrm{cert}}$.

C.4. Deletion Capacity of RR-DU and Comparison to DDP NetDP

We finally quantify the alignment bias introduced by RR-DU, its effect on utility, and how the resulting deletion capacity compares to the DDP NetDP baseline of Theorem 3.5. The privacy side (view-based DCU) follows the network-RDP amplification in Section A.1, adapted to the RR-DU routing pattern.

View-based amplification and DCU for RR-DU. Let u be the unlearning user. In RR-DU, at each unlearning round t the token is routed to u with probability p, and otherwise to a random non-unlearning user. Let M_u be the number of visits to u over T_u rounds; then $M_u \sim \mathrm{Binomial}(T_u, p)$ and $\mathbb{E}[M_u] = pT_u$. A Chernoff bound gives, for any $\beta \in (0, 1)$,

$$\mathbb{P}\big[\, M_u \ge (1 + \beta)pT_u \,\big] \;\le\; \exp\!\left(-\frac{\beta^2 pT_u}{3}\right).$$

Choosing β so that the right-hand side is at most $\delta/4$ ensures that with probability at least $1 - \delta/4$, the number of sensitive visits is at most a constant multiple of pT_u.

Condition on the event that $M_u \le c_0 pT_u$ for some constant c_0. Each visit to u corresponds to a Gaussian mechanism applied to a gradient of $\ell_{u\setminus f}$ (plus alignment terms), with ℓ_2-sensitivity proportional to L on the ball Θ_{cert}. On a complete

graph, the view-based amplification analysis of token-SGD in Lemma A.11, combined with the network-RDP definition (Definition A.7), implies that the contribution of a single such Gaussian step to the RDP at order α for the view of any other user $v \neq u$ satisfies

$$\rho_\alpha^{\text{view}} \lesssim \frac{\alpha L^2 \ln N}{\sigma^2 N}.$$

Composing over M_u visits gives

$$D_\alpha(Y_v \| Y_v') \lesssim \frac{\alpha L^2 \ln N}{\sigma^2 N} M_u \lesssim \frac{\alpha L^2 \ln N}{\sigma^2 N} pT_u,$$

with high probability over M_u. Using the standard RDP-to-DP conversion for Gaussian mechanisms [**?**] and Proposition A.13 then yields, up to constants,

$$\varepsilon \approx \frac{L}{\sigma}\sqrt{\frac{pT_u \ln N}{N \ln(1/\delta)}},$$

which is the (ε, δ)-network-DP guarantee quoted in the RR-DU view-based DCU result (Theorem 5.1 in the main paper). Applying Proposition C.1 shows that RR-DU achieves (ε, δ) decentralized certified unlearning on views for suitably calibrated σ.

Solving for σ gives the noise scale stated in the corresponding corollary:

$$\sigma = \Theta\left(\frac{L}{\varepsilon}\sqrt{\frac{pT_u \ln(1/\delta) \ln N}{N}}\right).$$

Alignment bias at the unlearning user. Next we quantify the bias introduced by the lightweight alignment step at user u.

Lemma C.7 (Alignment bias at the unlearning user)**.** *Let u be the unlearning user with local dataset $D_u = D_{u\setminus f} \cup D_f$ and $|D_f| = m$. Let $\mathcal{L}_{u\setminus f}(\theta)$ denote the empirical loss at u after deletion and let $\mathcal{L}_u(\theta)$ be the empirical loss before deletion. Under the gradient-boundedness assumption $\|\nabla_\theta \ell(\theta; z)\|_2 \leq L$ for all z, the discrepancy between the RR-DU update direction at u and the ideal negative gradient of $\mathcal{L}_{u\setminus f}$ satisfies*

$$\left\|\mathbb{E}\left[\Delta\theta_t \mid \theta_t\right]/\eta_t + \nabla_\theta \mathcal{L}_{u\setminus f}(\theta_t)\right\|_2 \leq C\frac{Lm}{n_u},$$

for some absolute constant C, where $\Delta\theta_t$ denotes the parameter update at a corrective step and $n_u = |D_u|$.

Proof. Write

$$\mathcal{L}_u(\theta) = \frac{1}{n_u}\sum_{z\in D_u} \ell(\theta; z), \qquad \mathcal{L}_{u\setminus f}(\theta) = \frac{1}{n_u - m}\sum_{z\in D_{u\setminus f}} \ell(\theta; z),$$

and

$$\mathcal{L}_f(\theta) = \frac{1}{m}\sum_{z\in D_f} \ell(\theta; z).$$

A simple algebraic identity gives

$$\nabla\mathcal{L}_u(\theta) = \frac{n_u - m}{n_u}\nabla\mathcal{L}_{u\setminus f}(\theta) + \frac{m}{n_u}\nabla\mathcal{L}_f(\theta).$$

In the EXACT mode, the corrective direction at a visit to u is $g_u = -\nabla\mathcal{L}_{u\setminus f}(\theta)$ (plus noise), so the bias is zero. In the LIGHTWEIGHT mode, the corrective step uses a minibatch $B_f \subseteq D_f$ scaled by m/n_u, which is an unbiased estimator of $(m/n_u)\nabla\mathcal{L}_f(\theta)$. The difference between this estimator and the ideal $-\nabla\mathcal{L}_{u\setminus f}(\theta)$ can be bounded by

$$\left\|\frac{m}{n_u}\nabla\mathcal{L}_f(\theta)\right\|_2 \leq \frac{m}{n_u}L,$$

using the per-example gradient bound. This yields the claimed $O(Lm/n_u)$ bound on the alignment bias, up to an absolute constant C. □

Deletion capacity of RR-DU and comparison with DDP NetDP. The RR-DU update at u thus decomposes into an ideal step on $\mathcal{L}_{u\setminus f}$ plus a small bias of order Lm/n_u, while updates at other users follow standard SGD on their local objectives. Combining Lemma C.7 with the utility bounds in Section C.5 and the DCU guarantee above, one obtains a two-regime deletion-capacity bound in which:

- For small m (so that Lm/n_u is negligible compared to the optimization and variance terms), the capacity of RR-DU scales similarly to the DDP NetDP baseline of Theorem 3.5, but with R replaced by R_{cert} and with an extra factor $\sqrt{p/s}$ in the effective variance: in this regime RR-DU essentially matches or improves on the DDP NetDP baseline due to its smaller certification domain and local averaging.
- For larger m, the alignment bias term $O(Lm/n_u)$ dominates and limits the admissible $m^\star$: beyond this point, increasing m would push the excess risk above the target γ, and the RR-DU capacity saturates faster than the DDP NetDP baseline, which does not suffer from this alignment bias but also does not perform targeted unlearning.

Qualitatively, RR-DU inherits the favorable $\sqrt{N}$-type amplification of the DDP NetDP baseline while operating on the smaller feasible region Θ_{cert} (diameter R_{cert}), and trades off targeted unlearning against an alignment bias that scales like m/n_u at the unlearning user.

C.5. Utility Bounds for RR-DU

We now sketch the optimization guarantees for RR-DU used in the main paper, building on the preliminaries above. We focus on the dependence on the unlearning horizon T_u, the local averaging factor s, and the effective variance.

Effective variance. For RR-DU, the noisy corrective steps occur only at the unlearning user u and only a fraction p of the time; non-unlearning users perform noiseless SGD with local averaging over s minibatches. A simple variance calculation shows that the effective variance entering the noisy-SGD analysis can be bounded as

$$G^2 \;\le\; L^2 \;+\; \frac{p}{s}\, d\sigma^2,$$

where we use that the gradient norms are bounded by L and that local averaging over s minibatches reduces the variance of the stochastic gradient noise by a factor $1/s$.

Strongly convex case. If $\mathcal{L}$ is μ-strongly convex and L-smooth over Θ_{cert}, a standard analysis of projected noisy SGD with decaying stepsize (under Assumptions C.3 and C.4) yields a last-iterate bound of the form

$$\mathbb{E}\big[\mathcal{L}(\theta_T) - \mathcal{L}(\theta^*)\big] \;\le\; \tilde{O}\!\left(\frac{L^2}{\mu s T_u} + \frac{pd\sigma^2}{\mu s T_u}\right),$$

where $\tilde{O}$ hides logarithmic factors, and the domain diameter enters through R_{cert} (absorbed into the constants).

Convex case. If $\mathcal{L}$ is convex and L-smooth over Θ_{cert}, but not strongly convex, projected noisy SGD with an appropriate stepsize schedule yields a last-iterate bound (via Corollary C.6 with $R_{\mathrm{dom}} = R_{\mathrm{cert}}$) of order

$$\mathbb{E}\big[\mathcal{L}(\theta_T) - \mathcal{L}(\theta^*)\big] \;\le\; \tilde{O}\left(\frac{R_{\mathrm{cert}} L}{\sqrt{s T_u}} + \sqrt{\frac{pd\sigma^2}{s T_u}}\right),$$

where the first term is the optimization error on the certification domain Θ_{cert} and the second term captures the variance.

Smooth non-convex case. If $\mathcal{L}$ is smooth but not necessarily convex on Θ_{cert}, the usual stationarity guarantees for noisy SGD give

$$\mathbb{E}\big[\|\nabla \mathcal{L}(\theta_{\mathrm{rand}})\|_2^2\big] \;\le\; \tilde{O}\!\left(\frac{L^2}{\sqrt{s T_u}} + \frac{pd\sigma^2}{\sqrt{s T_u}}\right),$$

for a randomly chosen iterate θ_{rand} from $\{\theta_t\}_{t=1}^{T_u}$.

D. Experimental Setup Details

This section provides additional details on the datasets, models, and hyperparameters used in the main experiments. Extended experimental settings are described in Appendix E.

D.1. Datasets

MNIST. We use the MNIST handwritten-digit dataset with $60\,000$ training and $10\,000$ test images of size 28×28 pixels (grayscale). All images are normalized to $[0, 1]$ by dividing pixel intensities by 255. For the main decentralized experiments, the training set is split i.i.d. and uniformly at random across $N = 10$ users so that each user receives $n_u = 6000$ examples.

CIFAR-10. For CIFAR-10 we use the standard $50\,000$ training and $10\,000$ test images of size 32×32 pixels with three color channels. Inputs are normalized channel-wise using the empirical training-set mean and standard deviation. We use standard augmentation: random horizontal flips and random crops with 4-pixel zero-padding. For the main experiments, the $50\,000$ training images are partitioned i.i.d. and uniformly across $N = 10$ users, giving $n_u = 5000$ examples per user. Appendix E additionally reports experiments with $N = 100$ and non-i.i.d. partitions.

Backdoor (BadNets) setup. To evaluate unlearning in the presence of backdoors, we consider a BadNets-type setting. A fixed trigger pattern (a small patch in the lower-right corner of the image) is overlaid on clean images, and all triggered images are relabeled to a single target class y^{bd}. In the main experiments on MNIST and CIFAR-10, we inject $m = 1000$ poisoned samples into the local dataset of a single *target user* u, which later issues the deletion request. During the training phase, we run Network-SGD (Algorithm 1) for $T = 100$ token hops over the graph. Backdoor accuracy is measured on a disjoint test set of triggered images, while clean accuracy is measured on the standard test sets without triggers. Appendix E additionally reports partial-deletion experiments parameterized by the ratio m/n_u, in which the number of deleted samples varies with the local dataset size of the target user.

Quantity	MNIST	CIFAR-10
# train examples	$60\,000$	$50\,000$
# test examples	$10\,000$	$10\,000$
Input size	28×28 (1 ch.)	32×32 (3 ch.)
Users N	10	10
Local train size n_u	6000	5000
Normalization	rescale to $[0, 1]$	channel-wise mean / std
Augmentation	none	flip + crop (4px padding)

Table 6. Dataset statistics and preprocessing for the main experiments.

D.2. Models

MNIST: FLNet. For MNIST we use a lightweight convolutional network (FLNet) with two convolutional layers, ReLU activations, 2×2 max-pooling, and a fully connected layer with softmax output. Batch normalization is applied after each convolution, and dropout (rate 0.5) is used before the final linear layer.

CIFAR-10: ResNet-18. For CIFAR-10 we adopt a standard ResNet-18 backbone with four residual stages of widths $64, 128, 256, 512$, batch normalization, and ReLU activations. The final fully connected layer produces the 10 logits before softmax. Global average pooling is applied before the last layer.

D.3. Training and Unlearning Hyperparameters

Unless otherwise specified, the main experiments share a common set of core hyperparameters for training and unlearning.

Trust-region and smoothness parameters. The trust-region radius ϱ and effective gradient bound L are tuned per dataset / model pair and then fixed for the corresponding setting in the main experiments.

Component	MNIST	CIFAR-10
Model	FLNet (2 conv + FC)	ResNet-18
Activation	ReLU	ReLU
Normalization	BatchNorm (conv layers)	BatchNorm (all residual blocks)
Pooling	2×2 max-pool	Global average pool (final)
Regularization	Dropout 0.5 (before FC)	none beyond standard ResNet-18
Output	10-way softmax	10-way softmax

Table 7. Model architectures (summary).

Hyperparameter	Value	Description
N	10	Number of users.
T	100	Training token hops (Network-SGD).
T_u	100	Unlearning token hops (RR-DU).
p	$1/N$	Routing probability toward unlearning user u.
s	4	Minibatches processed per token visit.
Optimizer	Adam	Used in the main experiments.
Learning rate η	0.005	Adam stepsize.
Minibatch size	64	Local batch size at each user.
(ε, δ)	$(1, 10^{-5})$	Target privacy for DDP / RR-DU capacity plots.

Table 8. Core training and unlearning hyperparameters for the main experiments.

Setting	ϱ	L
MNIST / FLNet	10.82	0.5
CIFAR-10 / ResNet-18	56.30	0.2

Table 9. Trust-region radius and smoothness surrogates used in the main experiments.

Baselines and noise calibration. The DDP baseline uses network-private SGD with projection onto a ball of radius $R = 10.0$ and gradient bound $L = 1.0$, which determine the Gaussian noise scale via the analysis in Section A.1. For DP-SGD we use per-example gradient clipping and Gaussian noise to match the same (ε, δ) level. The fine-tuning baseline simply re-optimizes on $D \setminus D_f$ starting from the pre-unlearning model.

Method	Key hyperparameters	Description
DDP (network-private SGD)	$R = 10.0$, $L = 1.0$	Projection radius and gradient bound for noise calibration.
DP-SGD	Clip $C = 5.0$	Per-example gradient clipping before adding Gaussian noise.
Fine-tuning	(no noise)	Retrain on $D \setminus D_f$ from pre-unlearning model.
RR-DU	(p, s, ϱ, σ)	Routing probability, local averaging, trust-region, noise scale.

Table 10. Baseline-specific hyperparameters for the main experiments (qualitative summary).

Backdoor-related parameters. In the main experiments, the forget set size at the target user is fixed to $m = 1000$ (the number of poisoned samples). The deletion capacity reported in the main paper is evaluated under this choice and the hyperparameters in Tables 8–10. Appendix E additionally reports experiments in which the deletion budget is parameterized by the ratio m/n_u.

Table 11. **Effect of trust-region radius ϱ in RR-DU (p=0.1).** "baseline" denotes the default value used in the main experiments. Results are from a single seed.

Dataset	ϱ	Clean Acc. (%)	Backdoor Acc. (ASR) (%)
MNIST	baseline	99.13	10.40
	5	99.12	15.60
	10	99.13	10.80
	25	99.04	10.20
	50	99.01	10.20
	60	99.01	10.20
CIFAR-10	baseline	88.12	12.00
	5	88.03	60.60
	10	88.41	46.40
	25	88.80	17.00
	50	88.12	12.00
	60	88.12	12.00

Table 12. **Effect of effective gradient bound L in RR-DU (p=0.1).** "baseline" denotes the dataset-specific default value used in the main experiments. Results are from a single seed.

Dataset	L	Clean Acc. (%)	Backdoor Acc. (ASR) (%)
MNIST	baseline	99.13	10.40
	0.1	99.20	10.20
	0.2	99.10	10.20
	0.5	99.13	10.40
	0.7	99.09	10.40
	1.0	99.10	10.40
CIFAR-10	baseline	88.12	12.00
	0.1	88.61	12.20
	0.2	88.12	12.00
	0.5	88.57	14.00
	0.7	87.34	16.60
	1.0	86.39	16.80

Table 13. **Effect of privacy parameter ε in RR-DU (p=0.1).** δ is fixed to 10^{-5}. Results are from a single seed.

Dataset	ε	Clean Acc. (%)	Backdoor Acc. (ASR) (%)
MNIST	0.1	99.19	60.60
	0.5	99.16	15.80
	2.0	99.12	10.20
	10	99.12	10.20
CIFAR-10	0.1	85.81	38.20
	0.5	88.66	14.00
	2.0	88.61	11.60
	10	89.06	10.80

Table 14. **Effect of privacy parameter δ in RR-DU (p=0.1).** ε is fixed to 1. Results are from a single seed.

Dataset	δ	Clean Acc. (%)	Backdoor Acc. (ASR) (%)
MNIST	10^{-4}	99.12	10.40
	10^{-6}	99.13	10.80
CIFAR-10	10^{-4}	89.05	12.00
	10^{-6}	88.68	11.20

E. Additional Experimental Results

E.1. Effect of Trust-Region Radius ϱ and Calibration Hyperparameters in RR-DU

We examine the sensitivity of **RR-DU** to the trust-region radius ϱ, the effective gradient bound L, and the privacy calibration parameters (ε, δ) on clean accuracy and backdoor accuracy (ASR) for MNIST and CIFAR-10. All runs use **RR-DU** with p=0.1 and a single seed.

Summary. Table 11 shows that the trust-region radius ϱ has a strong influence on backdoor accuracy (ASR), with a sub-

Table 15. **Effect of routing probability p in RR-DU (no projection, no noise).** p=0: fine-tuning (no targeted unlearning); p=1: *continuous unlearning* (always routing to the deleting client). Mean $\pm$ std over 3 seeds.

Dataset	p	Clean Acc. (%)	Backdoor Acc. (ASR) (%)
MNIST	0 (fine-tuning)	99.18 ± 0.04	10.33 ± 0.09
	0.1 (**RR-DU**)	99.18 ± 0.04	10.33 ± 0.09
	1.0 (continuous unlearning)	10.10 ± 0.50	0.00 ± 0.00
CIFAR-10	0 (fine-tuning)	88.61 ± 0.56	10.93 ± 0.25
	0.1 (**RR-DU**)	88.30 ± 0.35	11.20 ± 0.59
	1.0 (continuous unlearning)	17.74 ± 1.77	0.00 ± 0.00

Table 16. Aligned **RR-DU** exact-mode results on CIFAR-10 / ResNet18 (full topology, i.i.d. partition). The ratio m/n_u is reported only for partial-deletion settings; otherwise it is marked as N/A.

N	m/n_u	σ	Clean Acc.	Backdoor Acc. (ASR)	Wall-clock (s)
10	N/A	0.006	89.16	9.00	3441
100	N/A	0.000	83.87	7.40	2014
100	0.10	0.006	77.57 ± 0.90	0.00 ± 0.00	1384 ± 8

stantially larger effect on CIFAR-10 than on MNIST. In particular, very small radii lead to under-unlearning on CIFAR-10, whereas moderate to large radii reduce ASR to near the random-guessing regime with little change in clean accuracy. Table 12 indicates that the effective gradient bound L has a milder effect. On MNIST, performance remains nearly unchanged across the tested values, while on CIFAR-10 smaller values preserve performance close to the baseline and larger values ($L \geq 0.7$) lead to modest degradation in both clean accuracy and ASR. Table 13 confirms that ε is a key driver of unlearning quality: very small ε leaves substantial backdoor remnants, whereas $\varepsilon \geq 2$ reduces ASR to around 10–12% with little loss in clean accuracy. Finally, Table 14 shows that varying δ between 10^{-4} and 10^{-6} has negligible effect on either clean accuracy or ASR in this regime.

E.2. Effect of the Routing Probability p

Table 15 presents the effect of the routing probability p on clean accuracy and backdoor accuracy for MNIST and CIFAR-10.
Summary. Across MNIST and CIFAR-10, fine-tuning (p=0) and **RR-DU** with p=0.1 reach very similar final clean and backdoor accuracy. In contrast, p=1 (*continuous unlearning*) removes the backdoor completely but causes a severe collapse in clean accuracy, indicating over-unlearning. These results suggest that a small nonzero routing probability provides a favorable balance between targeted unlearning and retained utility in this noiseless ablation.

E.3. Additional Exact-Mode, Robustness, and Efficiency Results

We report several complementary results: aligned **RR-DU** exact-mode results with clean accuracy, backdoor accuracy (ASR), and wall-clock time; robustness to topology and scale on CIFAR-10 / ResNet18 under a non-i.i.d. partition; a full-topology exact sweep on CIFAR-10; a CIFAR-100 extension; and a limited matched efficiency comparison with PDUDT [**?**]. All timings are reported in wall-clock seconds. Mean $\pm$ std is reported only when multiple independent seeds are available.

Since the two protocols differ fundamentally—PDUDT relies on dynamic topologies and gossip, whereas **RR-DU** uses a random walk on a fixed graph—we do not view the comparison as fully head-to-head in general. We therefore report the matched comparison separately and interpret it cautiously.

Aligned exact-mode results. Table 16 reports aligned **RR-DU** exact-mode results on CIFAR-10 / ResNet18 under the full topology and an i.i.d. partition.
Summary. Across these aligned exact-mode runs, **RR-DU** maintains strong clean accuracy while reducing backdoor accuracy to a low level. In the partial-deletion setting, it achieves 0.00 ± 0.00 backdoor accuracy with 77.57 ± 0.90 clean accuracy.

Topology and scale robustness. For the non-i.i.d. experiments in Table 17, client datasets are generated using a class-wise Dirichlet partition with concentration parameter $\alpha = 0.5$, which induces moderate heterogeneity across users. We next evaluate exact-mode **RR-DU** under more challenging network topologies and larger client populations. Table 17 summarizes results on CIFAR-10 / ResNet18 for ring and grid topologies.

Table 17. Topology and scale robustness on CIFAR-10 / ResNet18 (**RR-DU**, exact mode, non-i.i.d. Dirichlet partition with $\alpha = 0.5$). All runs use σ=0.006.

Topology	N	Clean Acc.	Backdoor Acc. (ASR)	Wall-clock (s)
ring	10	73.08	12.20	1729
grid	10	70.16	11.00	1622
ring	100	69.60	34.60	1299
grid	100	69.32	38.00	1590

Table 18. Full-topology exact sweep on CIFAR-10 / ResNet18 (**RR-DU**, i.i.d. partition, σ=0.006). Mean $\pm$ std is reported over independent seeds; the last row includes two completed seeds.

N	m/n_u	Seeds	Clean Acc.	Backdoor Acc. (ASR)	Wall-clock (s)
10	0.05	3	81.90 $\pm$ 1.00	0.00 $\pm$ 0.00	1628 $\pm$ 19
10	0.10	3	81.90 $\pm$ 1.00	0.00 $\pm$ 0.00	1637 $\pm$ 16
100	0.05	3	75.97 $\pm$ 2.84	0.00 $\pm$ 0.00	1338 $\pm$ 19
100	0.10	2	77.57 $\pm$ 0.90	0.00 $\pm$ 0.00	1384 $\pm$ 8

Table 19. CIFAR-100 / ResNet-18 extension (**RR-DU**, exact mode; full topology, i.i.d. partition). Mean $\pm$ std is reported over three independent seeds.

N	m/n_u	Clean Acc.	Backdoor Acc. (ASR)	Wall-clock (s)
10	0.05	50.87 $\pm$ 1.21	1.20 $\pm$ 0.69	2524 $\pm$ 46
10	0.10	50.87 $\pm$ 1.21	1.20 $\pm$ 0.69	2460 $\pm$ 29
10	0.20	50.87 $\pm$ 1.21	1.20 $\pm$ 0.69	2470 $\pm$ 24
100	0.05	8.81 $\pm$ 0.44	0.93 $\pm$ 1.10	1915 $\pm$ 34
100	0.10	8.81 $\pm$ 0.44	0.93 $\pm$ 1.10	1903 $\pm$ 15
100	0.20	8.81 $\pm$ 0.44	0.93 $\pm$ 1.10	1952 $\pm$ 51

Table 20. **RR-DU** vs. PDUDT efficiency on matched CIFAR-10 / ResNet18 settings. All rows use N=10, σ=0.006, ring/grid topologies, and the same data partition as Table 17. PDUDT is substantially more expensive in wall-clock time, token hops, and communication volume.

Topology	Method	Clean Acc.	Backdoor Acc. (ASR)	Wall-clock (s)	Token Hops	Comm. Bytes	Wall Ratio
ring	RRDU	73.08	12.20	1729	100	4,469,584,800	1.0×
ring	PDUDT	45.02	0.00	56,548	3,800	169,844,222,400	32.7×
grid	RRDU	70.16	11.00	1622	100	4,469,584,800	1.0×
grid	PDUDT	45.35	0.00	65,244	5,000	223,479,240,000	40.2×

Summary. RR-DU continues to reduce backdoor accuracy on both ring and grid topologies. However, the non-i.i.d. setting becomes substantially more challenging at larger scale, resulting in lower clean accuracy and non-negligible residual backdoor accuracy relative to the i.i.d. full-topology setting.

Full-topology exact sweep on CIFAR-10. Table 18 reports the full-topology exact sweep on CIFAR-10 for different values of N and m/n_u.
Summary. Exact mode consistently reduces the backdoor accuracy to zero across all reported settings, while clean accuracy remains relatively stable as m/n_u increases from 0.05 to 0.10.

CIFAR-100 extension. We also evaluate exact-mode **RR-DU** on CIFAR-100 under the full topology and an i.i.d. partition. The results are reported in Table 19.
Summary. For N=10, the results are stable across the tested deletion ratios, with low backdoor accuracy and consistent clean accuracy. The N=100 setting is substantially more challenging in terms of clean utility, although backdoor accuracy remains low.

Matched efficiency comparison with PDUDT. Finally, Table 20 reports a limited matched comparison with PDUDT on CIFAR-10 / ResNet18 under ring and grid topologies.

Summary. In these matched runs, PDUDT removes the backdoor more aggressively, but at a much higher computational and communication cost. Its wall-clock time is $32.7\times$ higher on ring and $40.2\times$ higher on grid, together with substantially more token hops and communication. Since the underlying protocols differ, we view this table as an efficiency reference rather than a definitive head-to-head comparison.