An introduction to

Machine Unlearning

Amartya Sanyal

University of Copenhagen
Department of Computer Science

cope-forml.github.io

FoRML
Foundations of Responsible Machine Learning

Recall: differential privacy

Approximate DP
A randomized algorithm M is (ε, δ)-differentially private if for every pair of neighboring datasets S, S' and every measurable event E,

$$\Pr[M(S) \in E] \leq e^{\varepsilon} \Pr[M(S') \in E] + \delta.$$

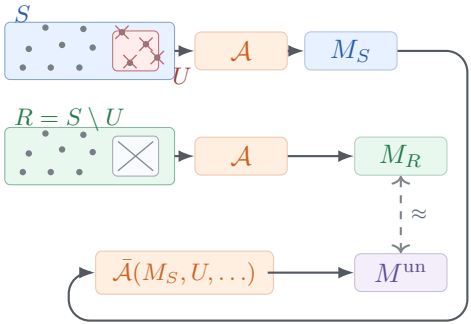
Today we will use the following notation to write this:

$$M(S) \approx_{\varepsilon,\delta} M(S')$$

What is the unlearning problem?

Notation

training set	$S = \{z_1, \ldots, z_n\}$
learning algorithm	$\mathcal{A}$
original model	$M_S = \mathcal{A}(S)$
training set	$S = \{z_1, \ldots, z_n\}$
learning algorithm	$\mathcal{A}$
original model	$M_S = \mathcal{A}(S)$
forget set	$U \subseteq S$
retain set	$R = S \setminus U$
retrained model	$M_R = \mathcal{A}(R)$
training set	$S = \{z_1, \ldots, z_n\}$
learning algorithm	$\mathcal{A}$
original model	$M_S = \mathcal{A}(S)$
forget set	$U \subseteq S$
retain set	$R = S \setminus U$
retrained model	$M_R = \mathcal{A}(R)$
unlearning algorithm	$\bar{\mathcal{A}}(M_S, U, \ldots)$
unlearned model	M^{un}



Exact and certified unlearning

Exact unlearning
For every dataset and unlearning request,

$$M^{\mathrm{un}} = M_R.$$

Certified approximate unlearning
Replace equality by indistinguishability,

$$M^{\mathrm{un}} \approx_{\varepsilon,\delta} M_R.$$

Cao and Yang 2015; Ginart et al. 2019

Cao and Yang 2015; Ginart et al. 2019; Guo et al. 2020

Example: participant withdrawal

The UK Biobank [79] holds genetic and medical data for half a million people. Thousands of ML models are trained on it. Thousands of papers use it.

Your paper might have used it.

Then one day you receive the following email:

```
EMAIL -- UK BIOBANK --
Subject:  UK Biobank Application [REDACTED], Participant Withdrawal Notification [REDACTED]

Dear Researcher,

As you are aware, participants are free to withdraw form the UK Biobank at any time and request that their
data no longer be used.  Since our last review, some participants involved with Application [REDACTED]
have requested that their data should longer be used.
```

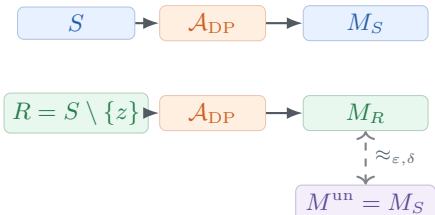
Differential privacy and unlearning

DP training gives a baseline certificate
If the training algorithm $\mathcal{A}$ is (ε, δ)-DP, then for a one-person withdrawal $R = S \setminus \{z\}$,

$$\mathcal{A}(S) \approx_{\varepsilon,\delta} \mathcal{A}(R).$$

Unlearning can be "do nothing"
Set $M^{\mathrm{un}} = M_S$. Then

$$M^{\mathrm{un}} = M_S \approx_{\varepsilon,\delta} M_R.$$



What this buys and costs
- Unlearning is fast.
- No need to store training data.

But:
- Cost is paid in training time and utility.
- Large deletion sets or repeated requests do not work with small ε.

Measuring utility in unlearning

Observation 1: Unlearning transfers utility. If $M^{\mathrm{un}} \approx_{\varepsilon,\delta} M_R$, then every utility event transfers:

$$\Pr\{F(M^{\mathrm{un}}) - F^\star > \alpha\} \le e^{\varepsilon} \Pr\{F(M_R) - F^\star > \alpha\} + \delta,$$
$$0 \le F - F^\star \le B \implies \quad \mathbb{E}[F(M^{\mathrm{un}}) - F^\star] \le e^{\varepsilon}\mathbb{E}[F(M_R) - F^\star] + \delta B.$$

Thus, if retraining has utility $\mathbb{E}[F(M_R) - F^\star] \le \gamma(|R|)$, then $\mathbb{E}[F(M^{\mathrm{un}}) - F^\star] \le e^{\varepsilon}\gamma(|R|) + \delta B$.

Observation 2a: empirical target.

$$\widehat{L}_R(w) = \frac{1}{|R|}\sum_{z\in R} \ell(w,z), \qquad \widehat{w}_R \in \arg\min_w \widehat{L}_R(w).$$

Competing with retraining on R is the natural benchmark for empirical-loss certificates.

Observation 2b: population target.

$$F(w) = \mathbb{E}_{z\sim\mathcal{D}}[\ell(w,z)].$$

The deleted points U are still samples from $\mathcal{D}$; exact retraining can leave statistical information on the table.

Deletion capacity. For an (ε,δ)-unlearning pair $(\mathcal{A},\bar{\mathcal{A}})$, with F the population loss,

$$m_{\varepsilon,\delta}^{\mathcal{A},\bar{\mathcal{A}}}(d,n) := \max\left\{ m : \ \mathbb{E}\left[\max_{\substack{U\subseteq S \\ |U|\le m}} F\Big(\bar{\mathcal{A}}(M_S, U, \ldots)\Big) - F^\star \right] \le 0.01 \right\}.$$

Sekhari, Acharya, Kamath, Suresh 2021

Two utility results

1. DP and unlearning separate.
DP hides every possible deletion *in advance*; unlearning can use the realized request U and stored statistics.

method	deletion capacity
DP / U-oblivious	$\widetilde{\Theta}\left(n/\sqrt{d}\right)$
unlearning, convex losses	$\widetilde{\Omega}\left(n/d^{1/4}\right)$

2. Pure unlearning can beat retraining.
For smooth strongly convex SCO, the near-optimal pure ε-unlearning excess-risk rate is

method	excess risk
RFS	$\frac{1}{n} + \left(\frac{m}{n}\right)^2$
near-optimal ε-unlearning	$\frac{1}{n} + \left(\frac{m}{n}\right)^2 e^{-2\varepsilon/(d+2)}$

If $\varepsilon \gg d$: exponential gain over retraining/DP.
If $\varepsilon \lesssim d$: retraining from scratch is optimal.

Sekhari, Acharya, Kamath, Suresh 2021; Regehr, Kamath, Lowy 2026

Exact unlearning

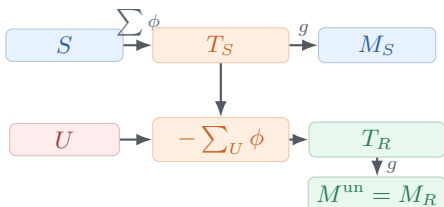
Exact unlearning from additive summaries

$\mathcal{A}$: train through a summary
Store an additive statistic and output the model:

$$T_S = \sum_{z \in S} \phi(z), \qquad M_S = g(T_S).$$

$\bar{\mathcal{A}}$: subtract the request
For $U \subseteq S$, update the stored state:

$$T_R = T_S - \sum_{z \in U} \phi(z), \qquad M^{\mathrm{un}} = g(T_R).$$



When it applies
- Count tables and histograms.
- Means and centroids.
- Covariance or Gram matrices.
- Naive Bayes.

Exact unlearning for ridge regression

$$w_S = \arg\min_w \tfrac{1}{2}\|X_S w - y_S\|_2^2 + \tfrac{\lambda}{2}\|w\|_2^2$$

$\mathcal{A}$: store normal equations

For ridge regression, keep

$$G_S = X_S^\top X_S + \lambda I, \qquad h_S = X_S^\top y_S,$$

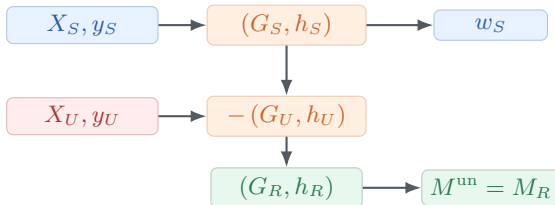
and return $M_S = w_S = G_S^{-1} h_S$.

$\bar{\mathcal{A}}$: delete by subtraction

For deleted rows X_U, y_U,

$$G_R = G_S - X_U^\top X_U, \qquad h_R = h_S - X_U^\top y_U,$$

$$M^{\mathrm{un}} = G_R^{-1} h_R = M_R.$$



Why this is exact

The retained normal equations are exactly (G_R, h_R), so solving them gives the same model as retraining on $R = S \setminus U$.

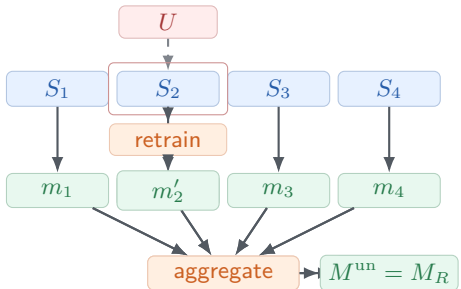
Exact unlearning by localizing retraining

$\mathcal{A}$: train many local models
Partition S into shards, train one model per shard, then aggregate:

$$M_S = \mathrm{Agg}(m_1, \ldots, m_K).$$

$\bar{\mathcal{A}}$: retrain only touched shards
If U lies in shard j, retrain that shard on its retained data and aggregate again:

$$M^{\mathrm{un}} = M_R.$$



SISA-style systems: Bourtoule et al. 2021

When additive summaries are not enough

- **Additive summaries work** when the computation is a stable function of the summary.
- **They become harder** when deletion changes latent assignments (clustering), optimization paths (SGD), tree structure (CART), or nonconvex basins (neural networks).

The problem with "close enough"

A parameter vector that is numerically close to retraining may still encode the removed point.

Certified Unlearning

Certified unlearning

Guo et al. certificate.

An unlearning mechanism M is certified if

$$M(\mathcal{A}(S), U) \approx_{\varepsilon,\delta} \mathcal{A}(S \setminus U).$$

Guo, Goldstein, Hannun, van der Maaten 2020

Certified data removal

Original optimisation problem

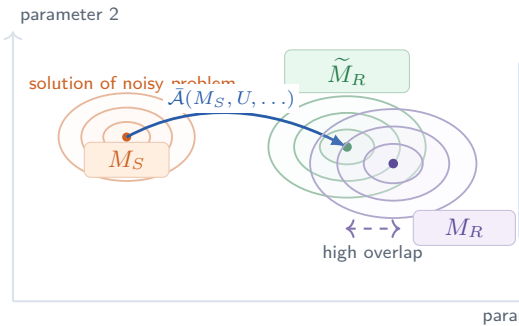
$$\min_{\theta} \sum_{z \in S} \ell(\theta; z) + \frac{\lambda}{2} \|\theta\|_2^2$$

add perturbation b

Noisy objective

$$\min_{\theta} \sum_{z \in S} \ell(\theta; z) + \frac{\lambda}{2} \|\theta\|_2^2 + b^\top \theta$$

$$P(b) \propto \exp(-\gamma \|b\|_2)$$



$\bar{A}(M_S, U, \ldots)$

1. $R \leftarrow S \setminus U$
2. $g_U \leftarrow \sum_{z \in U} \nabla \ell(M_S; z)$
3. $H_R \leftarrow \sum_{z \in R} \nabla^2 \ell(M_S; z) + \lambda I$
4. $\Delta\theta \leftarrow H_R^{-1} g_U$
5. return $\widetilde{M}_R \leftarrow M_S + \Delta\theta$

Proof follows Objective Perturbation style arguments

Certified data removal

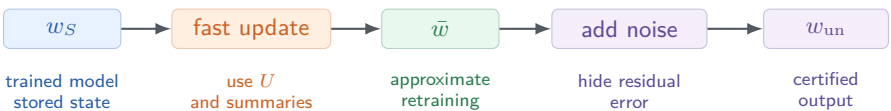
First clean certified unlearning algorithm for convex objectives.

- **Potentially cheaper than retraining.** When the Hessian H_R is available, unlearning is a Newton-style linear solve rather than a full optimization run.
- **Neat mechanism.** Objective perturbation randomizes the training objective; a Newton correction moves the trained model toward the retained-data optimum.
- **Drawbacks.**
  - Training must be altered, which lowers overall accuracy.
  - The method still needs access to the full training data to recompute the Hessian.
  - The exact unlearning parameter is data-dependent and hard to compute.

Guo, Goldstein, Hannun, van der Maaten 2020

A common unlearning template

The recurring pattern is to separate a **cheap update** from a **statistical certificate**.



- Train once and keep just enough state, often curvature or additive summaries.
- After a deletion request, compute a fast approximation $\bar{w}$ to retraining.
- Bound the residual, e.g. $\|\bar{w} - w_R\|$.
- Add calibrated noise so the residual cannot be statistically detected.

Algorithm 2: Newton unlearning for convex ERM

Learning algorithm: solve a regularized ERM problem

$$w_T = \arg\min_w \left[\frac{1}{|T|} \sum_{z \in T} \ell(w; z) + \frac{\lambda}{2} \|w\|_2^2 \right].$$

$\ell(w; z)$ is convex and smooth.

Newton correction

Input: w_S, deletion set U, retain set $R = S \setminus U$.

$g_U \leftarrow \sum_{z \in U} \nabla \ell(w_S; z)$.

$\widehat{H}_R \leftarrow \nabla^2 F_R(w_S)$ (or a stored curvature estimate).

$\bar{w} \leftarrow w_S + \frac{1}{|R|} \widehat{H}_R^{-1} g_U$.

return $\bar{w}$.

Add calibrated Gaussian noise: $w_{\text{un}} = \bar{w} + Z$.

Sekhari, Acharya, Kamath, Suresh 2021

Algorithm 2: How to calibrate the noise

Newton correction: $\bar{w} = w_S - H_R(w_S)^{-1}\nabla F_R(w_S)$

Goal

Bound the Newton residual $\|\bar{w} - w_R\|$, then add Gaussian noise Z calibrated to that bound.

Gradient at the old solution

Since $\nabla F_S(w_S) = 0$,

$$\nabla F_R(w_S) = -\frac{1}{|R|}\sum_{z\in U}\nabla \ell(w_S; z).$$

Taylor expansion around w_S

Retraining satisfies $\nabla F_R(w_R) = 0$. Taylor expand near w_S:

$$0 = \nabla F_R(w_R) = \nabla F_R(w_S) + H_R(w_S)(w_R - w_S) + r_T.$$

Solving the Taylor equation:

$$\begin{aligned} w_R &= \boxed{w_S - H_R(w_S)^{-1}\nabla F_R(w_S)} - H_R(w_S)^{-1} r_T \\ &= \boxed{\bar{w}} - \boxed{H_R(w_S)^{-1}} r_T. \end{aligned}$$

Residual after correction

$$r_T = \nabla F_R(w_R) - \nabla F_R(w_S) - H_R(w_S)(w_R - w_S).$$

With Hessian Lipschitz constant M, gradient bound L, and strong convexity λ,

$$\|\bar{w} - w_R\| \le \boxed{\lambda^{-1}}\, \|r_T\| \lesssim \frac{ML^2}{\lambda^3}\frac{|U|^2}{|S|^2} =: \rho_U.$$

$$\sigma \gtrsim \rho_U \frac{\sqrt{2\log(1.25/\delta)}}{\varepsilon}, \qquad Z \sim \mathcal{N}(0, \sigma^2 I_d).$$

$$w_{\mathrm{un}}(S, U) = \bar{w} + Z,$$
$$w_{\mathrm{un}}(S, U) \approx_{\varepsilon,\delta} w_{\mathrm{un}}(w_R, \emptyset).$$

Algorithm 3:Descent-to-Delete

Earlier convex methods can degrade across repeated unlearning; D2D studies sequential requests.

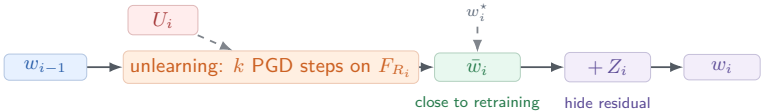
Learning target

$$w_i^\star = \arg\min_{w\in\Theta} F_{R_i}(w), \qquad F_{R_i}(w) = \frac{1}{|R_i|}\sum_{z\in R_i} \ell(w;z) + \frac{\lambda}{2}\|w\|_2^2.$$

$\ell(w;z)$ is convex, L-Lipschitz, and M-smooth.

Sequential deletion setting

- $R_i = R_{i-1} \setminus U_i$: the retained set after request i.
- $w_i^\star$: exact retraining solution on R_i.
- w_i: released unlearned model after repair and noise.



$$v_0 = w_{i-1}, \quad v_{t+1} = \Pi_\Theta\left(v_t - \eta\nabla F_{R_i}(v_t)\right), \quad t = 0,\dots,k-1,$$
$$\bar{w}_i = v_k, \qquad w_i = \bar{w}_i + Z_i.$$

Neel, Roth, Sharifi-Malvajerdi 2021; Hu et al., Online Learning and Unlearning

Descent-to-Delete: calibrating the noise

Repair step: $\Phi_R(w) = \Pi_\Theta(w - \eta \nabla F_R(w))$

Goal

Bound the distance from the repaired model w_k to retraining $w_R^\star$, then calibrate Gaussian noise.

Optima move slowly

If each loss is L-Lipschitz and $|U| = m$,

$$\left\| w_S^\star - w_R^\star \right\| \lesssim \frac{Lm}{\lambda n}.$$

D2D proof: optima move slowly

Stability claim

$$|S| = n,\ |U| = m,\ R = S \setminus U$$
$$\left\| w_S^\star - w_R^\star \right\| \lesssim \frac{Lm}{\lambda n}$$

1. Distance from first-order conditions

Let $d = w_S^\star - w_R^\star$. Strong convexity of F_R and optimality of $w_S^\star, w_R^\star$ give

$$\lambda \|d\|^2 \le \left\langle \nabla F_R(w_S^\star) - \nabla F_S(w_S^\star), d \right\rangle.$$

2. Cauchy–Schwarz

$$\|d\| \le \frac{\|\nabla F_R(w_S^\star) - \nabla F_S(w_S^\star)\|}{\lambda}.$$

3. Bound the gradient mismatch

Write $g_z = \nabla \ell(w_S^\star; z)$. The regularizer cancels in $\nabla F_R - \nabla F_S$:

$$\nabla F_R - \nabla F_S = \frac{m}{n(n-m)} \sum_{z \in R} g_z - \frac{1}{n} \sum_{z \in U} g_z.$$

Since ℓ is L-Lipschitz, $\|g_z\| \le L$:

$$\|\nabla F_R - \nabla F_S\| \le \frac{mL}{n} + \frac{mL}{n} = \frac{2Lm}{n}.$$

$$\left\| w_S^\star - w_R^\star \right\| \le \frac{2Lm}{\lambda n} \lesssim \frac{Lm}{\lambda n}.$$

Descent-to-Delete: calibrating the noise

Goal

Bound the distance from the repaired model w_k to retraining $w_R^\star$, then calibrate Gaussian noise.

Optima move slowly

If each loss is L-Lipschitz and $|U| = m$,

$$\left\| w_S^\star - w_R^\star \right\| \lesssim \frac{Lm}{\lambda n}.$$

GD contracts on the retained objective

For λ-strongly convex and M-smooth F_R,

$$\left\| \Phi_R(w) - w_R^\star \right\| \le \rho \left\| w - w_R^\star \right\|, \qquad \rho = \frac{M - \lambda}{M + \lambda} < 1.$$

D2D proof: GD contracts

Contraction claim

F_R is λ-strongly convex and M-smooth, $\eta = 2/(M+\lambda)$

$$\left\|\Phi_R(w) - w_R^\star\right\| \le \rho \left\|w - w_R^\star\right\|, \qquad \rho = \frac{M-\lambda}{M+\lambda}$$

1. Expand one descent step

Let $a = w - w_R^\star$ and $b = \nabla F_R(w) - \nabla F_R(w_R^\star)$.

$$\begin{aligned}\left\|\Phi_R(w) - w_R^\star\right\|^2 &\le \|a - \eta b\|^2 \\ &= \|a\|^2 - 2\eta\langle b, a\rangle + \eta^2 \|b\|^2 .\end{aligned}$$

2. Use smooth + strongly convex geometry

$$\langle b, a\rangle \ge \frac{\lambda M}{\lambda + M}\|a\|^2 + \frac{1}{\lambda + M}\|b\|^2 .$$

3. Substitute and choose η

$$\begin{aligned}\left\|\Phi_R(w) - w_R^\star\right\|^2 \le &\left(1 - \frac{2\eta\lambda M}{\lambda + M}\right)\|a\|^2 \\ &+ \eta\left(\eta - \frac{2}{\lambda + M}\right)\|b\|^2 .\end{aligned}$$

With $\eta = 2/(M+\lambda)$, the second term vanishes:

$$\left\|\Phi_R(w) - w_R^\star\right\|^2 \le \left(\frac{M-\lambda}{M+\lambda}\right)^2 \left\|w - w_R^\star\right\|^2 .$$

After k gradient steps,

$$\left\|w_k - w_R^\star\right\| \lesssim \rho^k \frac{Lm}{\lambda n}.$$

Descent-to-Delete: calibrating the noise

Repair step: $\Phi_R(w) = \Pi_\Theta(w - \eta \nabla F_R(w))$

Goal

Bound the distance from the repaired model w_k to retraining $w_R^\star$, then calibrate Gaussian noise.

Optima move slowly

If each loss is L-Lipschitz and $|U| = m$,

$$\left\| w_S^\star - w_R^\star \right\| \lesssim \frac{Lm}{\lambda n}.$$

GD contracts on the retained objective

For λ-strongly convex and M-smooth F_R,

$$\left\| \Phi_R(w) - w_R^\star \right\| \le \rho \left\| w - w_R^\star \right\|, \qquad \rho = \frac{M - \lambda}{M + \lambda} < 1.$$

Residual after k repair steps

Combining stability and contraction:

$$\left\| w_k - w_R^\star \right\| \lesssim \rho^k \frac{Lm}{\lambda n} =: \rho_U.$$

repair steps · deletion size · conditioning

Calibrate to the residual

Output $w_{\mathrm{un}} = w_k + Z$, where $Z \sim \mathcal{N}(0, \sigma^2 I_d)$.

$$\sigma \gtrsim \frac{\rho_U \sqrt{2 \log(1.25/\delta)}}{\varepsilon}.$$

What certified convex unlearning buys us

Certified unlearning works even when exact unlearning is unavailable.

- But the guarantees rely on strong assumptions, especially convexity.
- The added noise is calibrated to global parameters.

Can we go past these?

Unlearning beyond . . .

From here, proofs will get more informal.

Beyond convexity: a contraction template

We now look at two noisy algorithms that can yield certified guarantees for neural networks.

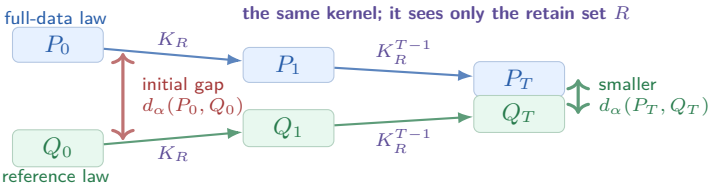
Notation. We use $d_\alpha(P, Q)$ for the α-Rényi divergence between distributions P and Q.

Central abstraction

A Markov kernel K maps a state x to a law $K(x, \cdot)$; PK is the output law when the input law is P. After deletion, both worlds receive the same retain-only kernel:

$$P_T = P_0 K_R^T, \qquad Q_T = Q_0 K_R^T.$$

The proof task is to show $d_\alpha(P_T, Q_T) \le c_T d_\alpha(P_0, Q_0)$ with $c_T \downarrow 0$.



Chien et al. 2024; Koloskova et al. 2025

Langevin unlearning: Algorithm

ERM and constraint:

$$f_S(\theta) := \frac{1}{n}\sum_{i=1}^{n} \ell(\theta; z_i), \qquad \mathcal{C}_B := \{\theta \in \mathbb{R}^d : \|\theta\|_2 \le B\},$$

where $\Pi_{\mathcal{C}_B}$ is Euclidean projection and τ^2 is the Langevin temperature/noise level.

Learning on the full data S

Projected noisy gradient descent (PNGD):

$$\theta_{t+1} = \Pi_{\mathcal{C}_B}\Big(\theta_t - \eta \nabla f_S(\theta_t) + \sqrt{2\eta\tau^2}\, Z_t\Big),$$
$$Z_t \sim \mathcal{N}(0, I_d).$$

Run the chain to stationarity:

$$\theta_\infty \sim \nu_S, \qquad \nu_S K_S = \nu_S.$$

Thus ν_S is the invariant law of the full-data Markov chain.

Unlearning on the retained data R

Let $R := S \setminus U$. Start from $\theta_0^{\mathrm{u}} \sim \nu_S$ and use only R:

$$\theta_{k+1}^{\mathrm{u}} = \Pi_{\mathcal{C}_B}\Big(\theta_k^{\mathrm{u}} - \eta \nabla f_R(\theta_k^{\mathrm{u}}) + \sqrt{2\eta\tau^2}\, \bar{Z}_k\Big),$$
$$\bar{Z}_k \sim \mathcal{N}(0, I_d).$$

The reference ν_R is the invariant law of the same chain run from scratch on the retained data.

Main question. Let $\rho_K := \mathrm{Law}(\theta_K^{\mathrm{u}})$. Can we prove that $d_\alpha(\rho_K, \nu_R)$ decreases with K? Here ρ_K is the unlearned output law and ν_R is the retraining law.

Langevin unlearning: informal guarantees

Theorem 3.1 — the target law exists
If $\mathcal{C}_B$ is compact with positive volume and ∇f_S is continuous, then PNGD has a unique invariant law ν_S, and $\mathrm{Law}(\theta_t) \Rightarrow \nu_S$ from every initial state.

Theorem 3.2 — privacy recuperation. Assume L-smoothness, G-Lipschitzness, and a log-Sobolev inequality for the relevant laws. If the initial stationary laws are $(\alpha, \varepsilon_{0,\alpha})$-RDP, then

$$d_\alpha(\nu_S, \nu_R) \le \varepsilon_{0,\alpha} \quad \Longrightarrow \quad d_\alpha\left(\mathrm{Law}(\theta_K^{\mathrm{u}}), \nu_R\right) \le \exp\left(-\frac{1}{\alpha}\sum_{k=0}^{K-1} r_k\right) \varepsilon_{0,\alpha}.$$

The positive rate r_k depends on the objective class.

General smooth nonconvex rate
Let $\rho_k = \mathrm{Law}(\theta_k^{\mathrm{u}})$, and let C_k be a constant that depends on ρ_k. Then

$$r_k = \log\left(1 + \frac{2\eta\tau^2}{(1+\eta L)^2 C_k}\right).$$

Projection and Gaussian smoothing give $C_k \le \widetilde{C}$, hence a uniform positive rate.

Gradient clipping + noisy finetuning: algorithm

Setup. The original learner $\mathcal{A}$ is arbitrary. The certificate is created after training by running noisy, clipped, retain-only updates:

$$M^{\mathrm{un}} = \bar{\mathcal{A}}(M_S, U, \ldots), \qquad \text{retrained output } = \bar{\mathcal{A}}(M_R, \emptyset, \ldots).$$

Both worlds use gradients from R only; no update accesses U.

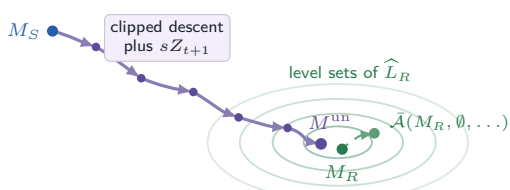
Gradient-clipping update

$$\theta_0 = \Pi_{C_0}(M_S), \qquad Z_{t+1} \sim \mathcal{N}(0, I_d),$$

$$\theta_{t+1} = \theta_t - \eta\Big(\Pi_{C_g}(g_t(\theta_t)) + \lambda_{\mathrm{reg}}\theta_t\Big) + sZ_{t+1}.$$

Here g_t is any stochastic gradient computed from R, and the output is $M^{\mathrm{un}} = \theta_T$.

Trajectory on the retained objective.



Gradient clipping + noisy finetuning: theorem

Assumptions
No convexity, smoothness, unique minimizer, or known Hessian bound. The proof uses clipping, Gaussian noise, and retained-data gradients.

Theorem 4.1, informally
For $0 < \varepsilon < 3\log(1/\delta)$, $\bar{\mathcal{A}}(M_S, U, \ldots) \approx_{\varepsilon,\delta} \bar{\mathcal{A}}(M_R, \emptyset, \ldots)$. It suffices to choose

$$\lambda_{\text{reg}} = 0: \quad s^2 = \frac{9\log(1/\delta)}{\varepsilon^2 T}\Big(C_0 + \eta C_g T\Big)^2.$$

If $1/2 < \eta\lambda_{\text{reg}} < 1$, it suffices that

$$s^2 = \frac{72\eta\lambda_{\text{reg}}\log(1/\delta)}{\varepsilon^2}\left(C_0(1-\eta\lambda_{\text{reg}})^T + \frac{C_g}{\lambda_{\text{reg}}}\right)^2.$$

Regularization dampens the initial-model term geometrically.

1. Couple retained runs
Compare the real run from M_S with the retain-only reference run from M_R. After initialization, both update using only R.

2. Clip controls drift
For $d_t = \|\theta_t - \theta'_t\|$, clipping gives

$$d_{t+1} \le |1 - \eta\lambda_{\text{reg}}| d_t + 2\eta C_g.$$

3. Noise pays for shifts
Gaussian noise hides a shift a at Rényi cost $qa^2/(2s^2)$. Optimizing the shift payments over T steps gives the stated noise.

Why global sensitivity is pessimistic

Global sensitivity asks for the largest adjacent change over *all* datasets:

$$\Delta_{\text{global}} = \sup_{S \sim S'} \|\mathcal{A}(S) - \mathcal{A}(S')\|.$$

DP-style calibration. Every neighboring world is treated as plausible, so the noise must hide the largest possible adjacent change.

Unlearning calibration. The retained set R is fixed by the deletion request; the certificate compares worlds that share R.

Unlearning should not pay for worst-case changes in retained data it is not trying to hide.

Heinzler, Malihi, Sanyal 2026

Privacy issues created by unlearning

Differencing attacks

Seeing M_S and M^{un}, an attacker can use

$$M_S - M^{\mathrm{un}}$$

to infer the deleted data U.

Chen et al. 2021

Protecting the undeleted

Even exact retraining can reveal facts about retained records across repeated requests.

Cohen, Kohen, Nissim, Stemmer 2026

Can the trained model unlearn by itself?

Every algorithm we saw kept deletion-relevant state.

Additive / ridge	sums, counts, $X^\top X$, $X^\top y$
SISA / shards	shard assignment, shard data, local models or checkpoints
Newton removal	retained data to form H_R, or a stored curvature estimate
D2D / Langevin	retained data for gradients, plus the current algorithmic state

Forget-only interface: at deletion time the unlearner receives only $M_S = \mathcal{A}(S)$ and the forget set U. No retained data, summaries, gradients, or checkpoints.

Natural question. If the trained model is the only stored state, can it support every future deletion request?

Teaching sets

A labelled sample T uniquely identifies h when

$$T \text{ teaches } h \iff \{g \in \mathcal{H} : g \text{ fits } T\} = \{h\},$$
$$\mathrm{TS}(h) := \min\{|T| : T \text{ teaches } h\}.$$

Define

$$N := N_{\mathrm{TD}}(\mathcal{H}, \mathsf{d}, \tau^\star, \eta)$$
$$= \mathrm{Pack}^{\mathsf{d}}_{2\eta}\left(\{h \in \mathcal{H} : \mathrm{TS}(h) \le \tau^\star\}\right).$$

Here N is the largest number of pairwise 2η-separated hypotheses, each teachable with at most $\tau^\star$ examples.

Informal theorem. Any "useful" forget-only unlearner for unlearning at most $\tau^{\star 2}$ must satisfy

$$\varepsilon \gtrsim \log N,$$

up to utility and approximation terms.

How much memorisation is needed?

Informal theorem (memorisation lower bound). Let $M_S = \mathcal{A}(S)$ and $W_i = \mathcal{A}(S \setminus U_i)$. For any ordering π of K deletion requests,

$$I(M_S; S) \gtrsim \underbrace{\sum_{i=1}^{K} H\left(W_{\pi(i)} \mid W_{\pi(<i)}, U_{\pi(\leq i)}\right)}_{\text{new information revealed by the retraining targets}} - \underbrace{K \operatorname{err}(\alpha, \varepsilon)}_{\text{slack from approximate unlearning}}.$$

Example: canonical thresholds. For deletion budget m, there are realizable threshold datasets of size n on domain of size N for which

$$I(M_S; S) \geq m \log\left(\frac{N}{n}\right) - \text{slack}.$$

Bare threshold ERM stores only $\log(N/n)$ bits: unlearning can require roughly m times more.

Informal theorem (eluder bound). For realizability testing, any learning–unlearning scheme for $\mathcal{H}$ needs $\Omega(\min\{n, \operatorname{Eluder}(\mathcal{H})\})$ bits.

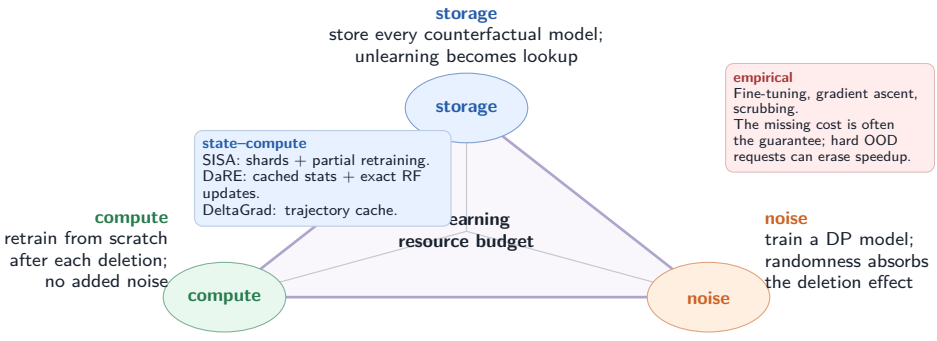
Radić, Singhal, Sanyal 2026; Cherapanamjeri et al. 2025

Newer topics

- **Source-free and retain-free unlearning:** reconstruct a proxy for R when raw retain data is unavailable.
- **Public-data assisted noise reduction:** use public/reference data to reduce the cost of noisy unlearning.
- **Distribution-shift unlearning:** non-i.i.d. unlearning requests can break assumptions behind convex certificates.
- **LLM and representation evaluation:** output-level forgetting can disagree with internal or paraphrased tests.

What is necessary for unlearning?

Unlearning is paid for with three resources.



Reading

1. Cao and Yang (2015); Ginart et al. (2019): exact unlearning and unlearning-efficient learning.
2. Guo et al. (2020): certified removal via Newton updates and perturbation.
3. Neel et al. (2021); Sekhari et al. (2021): certified convex unlearning.
4. Heinzler et al. (2026): retain sensitivity.
5. Chien et al. (2024); Koloskova et al. (2025): noisy dynamics and neural-network certificates.
6. Cherapanamjeri et al. (2025); Radić et al. (2026): memory and forget-only lower bounds.