CERTIFIED UNLEARNING DESPITE MILD REGULARIZATION AND NON-ISOTROPIC DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

We study machine unlearning: simulating the effect of the removal of a given set of samples without retraining from scratch. Existing methods for provable machine unlearning typically combine a fast approximate model update with noise calibrated to an upper bound on its distance from the ideal unlearned model. However, even for the relatively simple case of logistic regression, existing certified unlearning algorithms come at a severe cost to the model's predictive power, or trigger a retrain for a significant fraction of removal requests. We show that these limitations stem from a dependence on error bounds that require very strong regularization (resulting in a significant bias), and do not adapt to the geometry of the data (making the noise more detrimental to the model's predictions). We introduce UNCLE (Unlearning Certifiably via Local Estimates), a new algorithm for certified unlearning, which overcomes both of these issues. We obtain these improvements by applying a novel method for bounding the distance of any initial estimate from the ideal unlearned model, using only local properties of the loss landscape.

1 INTRODUCTION

As machine learning models become deeply integrated into everyday life, the ability to selectively remove the influence of individual training points has emerged as a fundamental requirement for several reasons. For example, a user may exercise a "right-to-be-forgotten" request, a curator may discover that some training samples were mislabeled, or a deployed system may need to remove poisoned or adversarial examples. The cleanest response is to retrain the model from scratch on the remaining data, but this is often computationally prohibitive. The goal of machine unlearning (Cao & Yang, 2015) is to obtain the same effect more efficiently: given a trained model and a deletion set T, update the model so that it behaves as if the samples in T had never been seen.

A promising approach to machine unlearning is to *approximate-and-noise*: one first computes a fast but potentially inaccurate approximation to the retrained model (e.g., using influence functions, gradient ascent, or a single Newton step), and noise is added to mask the difference between this approximation and a full retrain (e.g., Guo et al., 2020; Neel et al., 2021; Sekhari et al., 2021). This raises the natural question:

> *How should we calibrate our noise in order to effectively mask removed samples while preserving model utility?*

Many papers offer heuristic answers to this question (e.g., Graves et al., 2021; Jang et al., 2023; Goel et al., 2022; Kurmanji et al., 2023; Foster et al., 2024). While these heuristic methods allow for fast unlearning even for complex models, recent work has shown that they often fail to actually hide the unlearned samples (Bertran et al., 2024; Pawelczyk et al., 2025). Therefore, we turn to *certified* unlearning algorithms, which provably make the released model statistically indistinguishable from a retrained reference, using differential-privacy style guarantees (Ginart et al., 2019).

Despite strong theoretical results (Guo et al., 2020; Sekhari et al., 2021; Neel et al., 2021), these methods fall short in practice, even for the "simple" case of unlearning for logistic regression (Zou et al., 2025). We show in real-world datasets that they either come at a substantial reduction in accuracy, or else rely on triggering a full retrain on a significant fraction of removal requests. We demonstrate that these shortcomings are a direct result of the following weaknesses of existing

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

methods: a) over-reliance on strong regularization; and b) a lack of adaptation to the geometry imposed by the data.

Working in the simplified setting of logistic regression, we show that certified unlearning can be performed at a much milder cost to the model's accuracy by calibrating the scale of the noise to a tighter analysis of the fast approximation method's error and by drawing the noise from a carefully selected data-dependent geometry. Specifically, we introduce UNCLE (Unlearning Certifiably via Local Estimates) – a certified unlearning algorithm for logistic regression, which follows the same approximate-and-noise approach, but is able to achieve significantly better performance in practice.

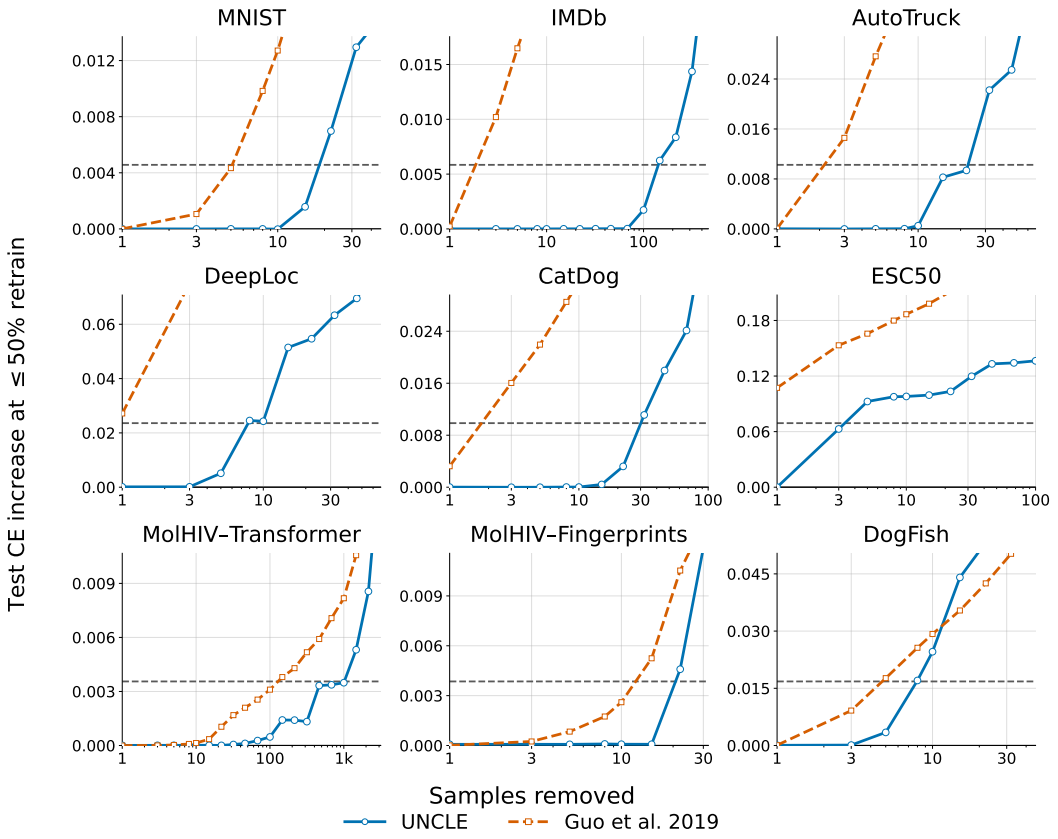


Figure 1: Unlearning algorithms using the approximate-and-noise paradigm trigger a full retrain when a Newton step approximation error exceeds the allowance determined by their noise scale. Scanning over hyperparameters of these algorithms, we report the lowest increase in test loss for which at most 50% of random removal requests trigger a full retrain, for fixed privacy parameters $\varepsilon = 1, \delta = 10^{-4}$. For interpretability of the significance of a given increase in test loss, for each dataset we plot (dashed horizontal line) the increase in test loss that corresponds to losing half of its samples at random (see Appendix C.6 for more details). We plot this increase in test loss as a function of the number of samples being removed for nine datasets representing a wide range of data types and modalities (see Table 1). For most of these datasets, the previous state-of-the-art of Guo et al. (2020) incurs a significantly higher cost to accuracy, to reach the same level of certification success.

1.1 Formal Definition

Let S be a dataset and let $\mathcal{M}$ be a training algorithm. We say that an *unlearning* procedure $\mathcal{U}$ successfully unlearns a set of samples $T \subseteq S$ from the output of $\mathcal{M}$, if the output of $\mathcal{U}$ is indistinguishable from an output that could have been generated using only the set of retained samples $R = S \setminus T$.

Definition 1 ((ε, δ)-Certified Machine Unlearning, Ginart et al. (2019))**.** *Fix $\varepsilon, \delta \geq 0$. The pair $(\mathcal{M}, \mathcal{U})$ is an (ε, δ)-certified machine unlearning algorithm if for every dataset S and every removal*

set $T \subseteq S$, denoting the retained set $R = S \setminus T$, we have

$$\mathcal{U}(\mathcal{M}(S), T) \approx_{\varepsilon,\delta} \mathcal{U}\left(\mathcal{M}(S \setminus T), \emptyset\right) = \mathcal{U}\left(\mathcal{M}(R), \emptyset\right) ,$$

in the sense that for any measurable event $\mathcal{E}$ on the output space,

$$e^{-\varepsilon}\Pr\left[\mathcal{U}\left(\mathcal{M}(R), \emptyset\right) \in \mathcal{E}\right] - e^{-\varepsilon}\delta \leq \Pr\left[\mathcal{U}(\mathcal{M}(S), T) \in \mathcal{E}\right] \leq e^{\varepsilon}\Pr\left[\mathcal{U}\left(\mathcal{M}(R), \emptyset\right) \in \mathcal{E}\right] + \delta .$$

Definition 1 is similar to that of differential privacy (Dwork et al., 2017), but instead of requiring privacy with respect to *every individual* datapoint in the input, unlearning requires privacy with respect to a *given subset* of the training set. Full retraining achieves $\varepsilon = 0, \delta = 0$, but the point of unlearning is to approximate this more efficiently – especially when $k = |T|$ is small.

1.2 Approximate-and-Noise

Many certified unlearning algorithms (Guo et al., 2020; Neel et al., 2021; Sekhari et al., 2021) follow an "approximate-and-noise" framework. Given the deletion set T, they first apply an *approximate unlearning* method: a method that computes an approximation $\hat{\boldsymbol{\theta}}_{\setminus T}$ to the ground-truth unlearned model $\boldsymbol{\theta}_{\setminus T}$ – the model that would have been produced by retraining from scratch. They then attempt to certify an upper bound on the distance between $\hat{\boldsymbol{\theta}}_{\setminus T}$ and the ground truth full retrain $\boldsymbol{\theta}_{\setminus T}$.

If the certification succeeds, this means that the approximation is close to the ground truth in model space, and this proximity is converted to an unlearning guarantee via a masking procedure which either adds noise directly to the unlearned model weights (Neel et al., 2021; Sekhari et al., 2021; Pandey et al., 2026) or applies an objective perturbation (Guo et al., 2020).

Algorithm 1: Approximate-and-Noise Unlearning

Input: Dataset S, deletion set $T \subseteq S$, trained model $\boldsymbol{\theta}$, radius $r > 0$, parameters (ε, δ)

$\hat{\boldsymbol{\theta}}_{\setminus T} \leftarrow$ `ApproximateUnlearning`$(S, T, \boldsymbol{\theta})$

if *not* `CertifyBound`$(S \setminus T, \hat{\boldsymbol{\theta}}_{\setminus T}, r)$ **then**

  $\hat{\boldsymbol{\theta}}_{\setminus T} \leftarrow$ `Retrain`$(S \setminus T)$

return `AddNoise`$(\hat{\boldsymbol{\theta}}_{\setminus T}, r, \varepsilon, \delta)$

Therefore, an unlearning algorithm within this paradigm requires three components: an approximate unlearning procedure that is both fast and accurate, a procedure that certifies tight bounds on its approximation error, and some way to mask this approximation error.

1.3 Our Contributions

We improve both the CertifyBound and AddNoise steps, and our methods can be applied with any ApproximateUnlearn procedure. In UNCLE, we adopt the single Newton step method used in Guo et al. (2020); Sekhari et al. (2021), for its efficiency and accuracy for typical datasets (Koh et al., 2019; Rubinstein & Hopkins, 2025; 2026)[1].

$$\text{NewtonStep} \coloneqq \hat{\boldsymbol{\theta}} - \left[\nabla^2 L_R|_{\hat{\boldsymbol{\theta}}}\right]^{-1} \nabla L_R|_{\hat{\boldsymbol{\theta}}} \approx \hat{\boldsymbol{\theta}}_R \coloneqq \operatorname*{argmin}_{\boldsymbol{\theta} \in \mathbb{R}^d} \{L_R(\boldsymbol{\theta})\} .$$

CertifyBound: Unlearning Despite Mild Regularization Our main contribution is improving this step of the approximate-and-noise paradigm. Current approximate-and-noise approaches (Guo et al., 2020; Neel et al., 2021; Sekhari et al., 2021) are based on certificates that require global strong convexity and smoothness conditions on the loss function (Rahnama Rad & Maleki, 2020; Koh et al., 2019; Giordano et al., 2019; Wilson et al., 2020; Zou et al., 2025).

However, logistic regressions, are not *globally* strongly convex. This is because in the limit of the model weights going to infinity, the minimum eigenvalue of the Hessian of the underlying loss goes

[1] Some recent literature is pessimistic about the use of a single Newton step for unlearning and proposes using multiple Newton steps or relaxing the conditions of unlearning (Zou et al., 2025; Pandey et al., 2026).

to 0. This might seem like a technicality, but because existing unlearning algorithms rely on global strong convexity of the loss, they require a large strongly-convex regularizer when the loss is not globally strongly convex. This comes at a cost: too much regularization substantially degrades test accuracy of the learned model by introducing significant bias.

$$\inf_{\boldsymbol{\theta}\in\mathbb{R}^d}\left\{\sigma_{\min}\left(\nabla^2 L|_{\boldsymbol{\theta}}\right)\right\} = \lim_{\boldsymbol{\theta}\to\infty}\left\{\sigma_{\min}\left(\nabla^2 L|_{\boldsymbol{\theta}}\right)\right\} = \underbrace{\lambda}_{\substack{\text{Regularization}\\ \text{Coefficient}}}$$

More concretely, because these bounds on the Newton step approximation error scale inverse-polynomially with the global strong convexity, unlearning algorithms based on these analyses are forced to either increase their regularization or add noise that scales poorly with the native regularization.

$$\rho_{\text{previous}} \propto \frac{1}{\lambda^c}\,, \quad \text{for } c \geq 2\,.$$

As observed by Koh et al. (2019), this represents a significant gap between the empirical observation that the single Newton step is usually accurate in practice but theoretical guarantees remain loose:

> (Koh et al., 2019): $\lambda/\sigma_{\max}$ is quite small in our experiments [...] so the actual correlation of influence is better than predicted by this theory.

Rubinstein & Hopkins (2026) partially resolve this theoretical gap by deriving approximation-error bounds that depend on *local* curvature, which they show to be much larger than the typical regularization selected by cross-validation (CV) for logistic regressions with "well-behaved" covariates.

$$\mathbf{H}_{\text{train}} \coloneqq \nabla^2 L|_{\boldsymbol{\theta}} \approx \mathbf{X}^\intercal\mathbf{X} + \lambda\mathbf{I} \gg \lambda\mathbf{I}\,.$$

However, Rubinstein & Hopkins (2026)'s bounds do not translate into a CERTIFYBOUND-style algorithm, because their bounds are not *efficiently verifiable*. In other words, while Rubinstein & Hopkins (2026) prove that a single Newton step is accurate for a well-behaved dataset, they do not provide an efficient algorithm that maps a specific dataset / deletion request to a concrete numerical bound on the Newton step approximation error. This is not an existing capability that Rubinstein & Hopkins (2026) fail to report: in Appendix E, we show that applying (Rubinstein & Hopkins, 2026) in such a manner seems to require solving computational problems widely believed to be intractable.

At a high level, the difficulty with applying (Rubinstein & Hopkins, 2026) algorithmically stems from their requirement that local conditions hold within an ellipsoid, and at the core of our certification step lies the observation that it is much easier to verify these local conditions when we are allowed to also add linear inequality constraints to the area where these conditions are required (see Figure 2).

We introduce a novel proof technique to show that any convex set suffices (such as the intersection of an ellipsoid with linear inequality constraints). By efficiently instantiating a local-conditions-bound, we are able to calibrate the scale of noise necessary for certified unlearning much more accurately than previous approaches (Guo et al., 2020; Sekhari et al., 2021; Pandey et al., 2026) and without appealing to very high regularization coefficients, resulting in a much lower loss of accuracy due to the unlearning process.

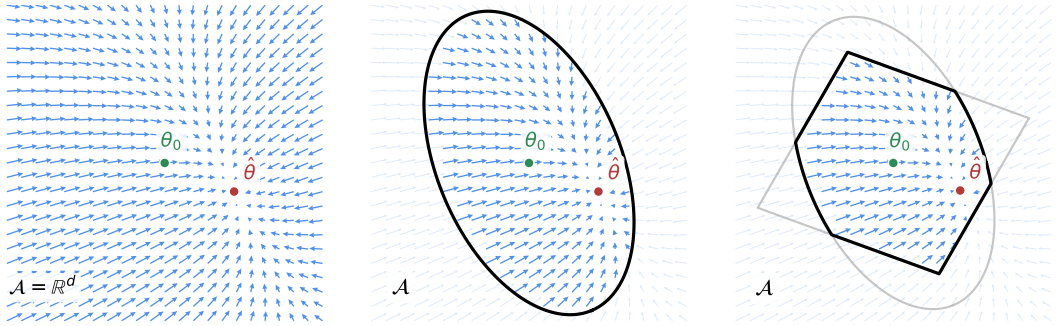
The following Theorem 1.1 generalizes the main result of (Rubinstein & Hopkins, 2026), who proved a similar result for the special case when $\mathcal{A}$ is an ellipsoid.

Theorem 1.1 (Generalization of Theorem 1.1 of (Rubinstein & Hopkins, 2026) – Informal, see Theorem D.1)**.** *For any open, bounded, convex neighborhood $\mathcal{A}$, any point $\boldsymbol{\theta}_0 \in \mathcal{A}$ and any loss function L that is smooth and strongly convex in $\mathcal{A}$, if L also satisfies that*

$$\forall \boldsymbol{\theta} \in \mathcal{A},\ \boldsymbol{\theta}_0 - \left(\nabla^2 L|_{\boldsymbol{\theta}}\right)^{-1} \nabla L|_{\boldsymbol{\theta}_0} \in \mathcal{A}\,,$$

then $\mathcal{A}$ contains a local minimum $\hat{\boldsymbol{\theta}}$ of the loss L.

This generalization is required for the design of our UNCLE algorithm, but necessitates a new proof technique, which we present Appendix D.



(a) Conditions must hold everywhere.

(b) Rubinstein & Hopkins (2026): it suffices to check conditions in a "sufficiently large" ellipsoid.

(c) Theorem 1.1: any "sufficiently large" convex set suffices.

Figure 2: Visual depiction of progress in theoretical analyses of unlearning algorithms. **(a)** Until recently, the leading analyses (Rahnama Rad & Maleki, 2020; Giordano et al., 2019; Koh et al., 2019; Wilson et al., 2020; Zou et al., 2025) required global conditions on the loss, necessitating a very large regularization coefficient for their guarantees to apply. **(b)** Rubinstein & Hopkins (2026) showed that it suffices for these conditions to hold within a relatively small ellipsoid and that these local conditions hold for "well-behaved" regression problems. However, turning this result into a certified unlearning algorithm requires some way of verifying these conditions for a specific dataset. **(c)** In Theorem 1.1, we show that it suffices for these conditions to hold in any "sufficiently large" convex set $\mathcal{A}$. This seemingly minor difference enables our UNCLE algorithm, which applies Theorem 1.1 to a more complex set, and it also requires a fundamentally different proof technique.

By obtaining this tighter bound on the error of an approximate unlearning algorithm, we can obtain a certified unlearning algorithm whose noise is calibrated to the right *scale* without appealing to the very strong regularization needed to apply the previous analyses (Rahnama Rad & Maleki, 2020; Giordano et al., 2019; Koh et al., 2019; Wilson et al., 2020).

ADDNOISE: Calibrating Noise Geometry Most feature matrices in our benchmark were non-isotropic and more than that, had high condition numbers (see κ in Table 1). This motivates the need for an unlearning algorithm that is calibrated to the geometry of the problem, which is our second contribution.

The effects of noise on our models' accuracy are best understood by considering a second order Taylor expansion of the loss when the model weights $\boldsymbol{\theta}$ are perturbed by a mean-$\mathbf{0}$ random vector $\boldsymbol{\nu}$:

$$L_{\text{test}}(\boldsymbol{\theta}+\boldsymbol{\nu}) \approx L_{\text{test}}(\boldsymbol{\theta}) + \underbrace{\langle \boldsymbol{\nabla} L_{\text{test}}|_{\boldsymbol{\theta}}, \boldsymbol{\nu}\rangle}_{\text{small, mean } \mathbf{0}} + \frac{1}{2}\boldsymbol{\nu}^{\intercal}\left[\boldsymbol{\nabla}^2 L_{\text{test}}|_{\boldsymbol{\theta}}\right]\boldsymbol{\nu} + \text{higher-order terms}$$

For a logistic regression, the geometry of this second order effect typically resembles the geometry of the covariates themselves, and so does the geometry of the Newton step error itself (Rubinstein & Hopkins, 2026).

$$\mathbf{H}_{\text{test}} \coloneqq \boldsymbol{\nabla}^2 L|_{\boldsymbol{\theta}} \preceq \boldsymbol{\Sigma}_{\text{test}} \coloneqq \mathbf{X}_{\text{test}}^{\intercal}\mathbf{X}_{\text{test}} \approx \boldsymbol{\Sigma}_{\text{train}}\,.$$

For this reason, we set the noising step of our UNCLE algorithm to add Gaussian noise drawn from the dual geometry $\nu \sim \mathcal{N}\left(\mathbf{0}, \approx \rho \cdot \boldsymbol{\Sigma}^{-1}\right)$ (where ρ represents the scale of the noise), which results in only a relatively mild increase to out-of-sample loss[2].

$$\mathop{\mathbb{E}}_{\nu}\left[L(\boldsymbol{\theta}+\boldsymbol{\nu}) - L(\boldsymbol{\theta})\right] \approx \frac{1}{2}\mathop{\mathbb{E}}_{\nu}\left[\nu^{\intercal}\mathbf{H}_{\text{test}}\nu\right] \lesssim \frac{\rho d}{2}\,.$$

[2]In practice, $\boldsymbol{\Sigma} = \mathbf{X}^{\intercal}\mathbf{X}$ is often singular or ill-conditioned (see Table 1), so the UNCLE algorithm adds noise in a mildly regularized geometry $\mathbf{Q} = \mathbf{X}^{\intercal}\mathbf{X} + \tau\mathbf{I}$.

By contrast, many unlearning approaches add isotropic Gaussian noise (Sekhari et al., 2021; Neel et al., 2021; Pandey et al., 2026). However, as was shown for the case of private linear regression (see e.g., Brown et al. (2024); Anderson et al. (2025); Rubinstein et al. (2026)), adding isotropic noise to linear models fit on non-isotropic data can cause the loss in accuracy to grow polynomially with the condition number of the covariates.

Guo et al. (2020) use a different approach where they add an objective perturbation to their training loss. To do this, they draw an isotropic Gaussian perturbation $\mathbf{b} \sim \mathcal{N}(\mathbf{0}, \text{scale}' \cdot \mathbf{I})$ and perturb their loss function:

$$L(\boldsymbol{\theta}) \to L'(\boldsymbol{\theta}) \coloneqq L(\boldsymbol{\theta}) + \langle \boldsymbol{\theta}, \mathbf{b} \rangle .$$

To estimate the effect this has on the trained model, consider what a single Newton step update would predict for the effect of this perturbation:

$$\boldsymbol{\theta}' \approx \boldsymbol{\theta} - \mathbf{H}_{\text{train}}^{-1} \mathbf{b} .$$

In other words, this loss perturbation strategy would again be distributed according to the wrong geometry, and we therefore do not expect it to adapt well to datasets with high condition numbers.

$$\boldsymbol{\theta}' - \boldsymbol{\theta} \approx -\mathbf{H}_{\text{train}}^{-1} \mathbf{b} \sim \mathcal{N}(\mathbf{0}, \rho' \cdot \mathbf{H}_{\text{train}}^{-2}) \approx \mathcal{N}(\mathbf{0}, \rho' \cdot \boldsymbol{\Sigma}^{-2})$$

We note that in many differential privacy settings, making the noise adapt to the data geometry is a challenge in and of itself, because the noise itself can leak information about the training set (see e.g., (Brown et al., 2023; 2024)). A key observation is that this is no longer an issue when we switch to unlearning, where we are allowed to draw our noise from any distribution, as long as that distribution depends only on the retained samples.

The UNCLE Algorithm Combining these components into the approximate-and-noise framework, we get UNCLE– Algorithm 2.

Algorithm 2: UNCLE: certified unlearning for logistic regression. The key technical component is the FOUNDCERTIFICATE step which certifies the approximation error bound using Theorem 1.1 *without performing a new optimization.*

Input: Data $(\mathbf{x}_i, Y_i) \in \mathbb{R}^d \times \{0, 1\}$ for $i \in [n]$; regularization coefficient $\lambda > 0$; initial model $\theta_0 \in \mathbb{R}^d$; deletion set $T \subseteq S = [n]$; slack radius $R_2 > 0$; masking regularization coefficient $\tau > 0$; privacy parameters $\varepsilon, \delta \in \mathbb{R}^+$.

Output: An unlearned model $\boldsymbol{\theta}' \in \mathbb{R}^d$.

Notation: $R = S \setminus T$, $L_R(\boldsymbol{\theta}) = \sum_{i \in R} \ell(\mathbf{x}_i^\intercal \boldsymbol{\theta}, Y_i) + \frac{\lambda}{2} \|\boldsymbol{\theta}\|_2^2$, and $\mathbf{Q}_\tau = \mathbf{X}_R^\intercal \mathbf{X}_R + \tau \mathbf{I}_d$.

$\hat{\boldsymbol{\theta}} \leftarrow$ `NewtonStepUpdate` $(\boldsymbol{\theta}_0, X_R, Y_R; \ell, \lambda)$

if *not* `FoundCertificate` $\left(\left\| \mathbf{Q}_\tau^{1/2} \left(\hat{\boldsymbol{\theta}} - \operatorname{argmin}_{\boldsymbol{\theta} \in \mathbb{R}^d} \{ L_R(\boldsymbol{\theta}) \} \right) \right\|_2 \leq R_2 \right)$ **then**

  $\hat{\boldsymbol{\theta}} \leftarrow \operatorname{argmin}_{\boldsymbol{\theta} \in \mathbb{R}^d} \{ L_R(\boldsymbol{\theta}) \}$ `// Full retraining`

$\boldsymbol{\nu} \sim \mathcal{N}\left(0, 2 \frac{\log(1.25/\delta)}{\varepsilon^2} R_2^2 \mathbf{Q}_\tau^{-1}\right)$

return $\hat{\boldsymbol{\theta}} + \boldsymbol{\nu}$

We describe the key subroutine `FoundCertificate` in Appendix B, where we also prove:

Theorem 1.2 (See Theorem B.2)**.** UNCLE *(Algorithm 2) is an* (ε, δ)*-unlearning algorithm.*

1.4 EXPERIMENTAL DESIGN

We evaluate the behavior of UNCLE empirically and compare its unlearning capabilities to (Guo et al., 2020) which represents the previous state-of-the-art in our setting (see related works section). We measure the performance of both algorithms on random deletion requests for a suite of 9 real-world logistic regression datasets (see Table 1). We fix relatively mild privacy parameters ($\varepsilon = 1$, $\delta = 10^{-4}$) and evaluate the unlearning algorithm on two axes: runtime and accuracy.

Table 1: Dataset statistics (n is the number of training samples, d is the number of parameters, accuracy is the test accuracy in % with no unlearning, κ is the condition number of $\mathbf{X}^T\mathbf{X}$ where $\mathbf{X}$ is the design matrix). We then report training time and query times for each unlearning algorithm (query times represent Newton step + certification + masking time measured for deletion requests of size $k = 100$, and exclude setup and full-retrain fallbacks), cross-entropy thresholds (τ_{CE}) and deletion capacity. All timings are in milliseconds and were measured on Google Cloud `c3-highmem-22` VM with Intel Xeon Platinum 8481C processors. CE thresholds are defined by the mean increase in test cross-entropy due to a loss of 50% of samples at random (see Appendix C.6). Deletion capacity ($k^\star$) is the largest tested value of k for which a retrain rate of $\leq 50\%$ and test CE increase below the threshold were simultaneously achieved. We report all datasets tested in our study.

					Time (ms)				Deletion capacity	
Dataset	n	d	Acc.	κ	t_{train}	$t_{\text{Guo et al.}}$	t_{UNCLE}	τ_{CE}	$k^\star_{\text{Guo et al.}}$	$k^\star_{\text{UNCLE}}$
MNIST	6,000	784	96.88	∞	64	5.6	13	0.0046	5	**15**
IMDb	40,000	769	87.74	2.3e16	5,200	6.7	33	0.0058	1	**100**
AutoTruck	9,600	513	93.08	5.1e5	250	4.5	7.8	0.010	1	**22**
DeepLoc	5,864	641	90.39	3.1e17	390	6.5	8.2	0.024	0	**5**
CatDog	9,600	513	82.79	3.3e5	320	4.5	8.1	0.0099	1	**22**
ESC50	1,600	513	87.50	2.5e7	110	4.4	4.8	0.069	0	**3**
MH–T	32,901	769	96.94	9.0e16	1,800	5.3	25	0.0036	100	**1e3**
MH–F	32,901	2,049	97.13	6.4e3	1,600	48	92	0.0039	10	**15**
DogFish	1,800	2,048	98.67	∞	30	86	140	0.017	3	**5**

Runtime / retrain rate. In practice, the Newton step + certification steps of the unlearning algorithms were significantly faster than a full retrain across all but one dataset. Therefore, the runtime of these approximate-and-noise unlearning algorithms is effectively controlled by their retrain rate – the fraction of unlearning requests that are serviced by triggering a full retrain.

In Figure 1 we set a 50% retrain budget, and report on smallest increase to test cross-entropy achieved by each algorithm within this budget.

Test accuracy / loss Most unlearning algorithms come at a cost to the predictive performance of the models, both through the need to add noise (for masking purposes) and through an increased regularization which introduces additional bias to the model.

Sekhari et al. (2021) set an increase of 0.01 in mean test loss as a threshold, meant to represent a small constant loss of accuracy, but do not justify the selection of this specific constant. To obtain a more calibrated scale for what might represent an "acceptable" loss of accuracy for a model, we use the expected loss of accuracy due to losing a given fraction of the training data (see Appendix C.6). In particular, we measure the mean increase in test loss if 50% of the training data are deleted at random, and use this as a relatively lax threshold for an acceptable increase in test loss.

Sekhari et al. (2021) define the *deletion capacity* of a dataset as the maximum size of the unlearning request that can be serviced given a fixed budget for test loss increase[3]. We further require that at most 50% of removal requests trigger a full retrain.

Hyper-parameter scan. Unlearning algorithms often require additional hyper-parameters, in addition to those used by the underlying learning problem. In particular, both Guo et al. (2020) and UNCLE require a scale for their masking noise (decreasing the noise reduces the negative effects on model accuracy but also increases the fraction of requests that must be serviced via a full retrain), and both also require an increased regularization coefficient (higher regularization allows us to add less noise, but also introduces a bias).

[3] Sekhari et al. (2021) suggest a budget of 0.01 increase in test loss. We support using this target since we observed that a 0.01 test loss increase is roughly comparable to losing a significant fraction of samples in one's training set.

Existing literature does not give a clear formula for how one should pick the hyper-parameters of their unlearning algorithm, so to give algorithms a fair comparison, we perform a careful scan for the best hyper-parameter values, and report on the behavior of the best hyperparameters that satisfy the relevant budget (retrain rate or test loss increase), evaluating the chosen hyper-parameters with multiple fresh seeds to produce confidence intervals (see Appendix C for more details).

1.5 Related Work

The removal of specific training samples from a machine learning model has been studied in prior work (Cao & Yang, 2015; Ginart et al., 2019; Bourtoule et al., 2021). Within machine unlearning, *certified* unlearning requires formal guarantees that an updated model is statistically indistinguishable from one retrained without the deleted data (Guo et al., 2020; Neel et al., 2021; Sekhari et al., 2021; Koloskova et al., 2025). We focus on algorithms that satisfy these strong certified unlearning guarantees. We do not include heuristic approaches (see Section 1) or approaches with different or weaker guarantees (Mahadevan & Mathioudakis, 2022; Gunn, 2026; Pandey et al., 2026).

We evaluate UNCLE on a wide range of real-world datasets and compare it to Guo et al. (2020), which to the best of our knowledge represents the previous state of the art for logistic regression unlearning. This baseline does not cover all previous unlearning algorithms. In Appendix A, we explore several additional certified unlearning algorithms, the challenges of integrating them as baselines, and why a comprehensive benchmark remains open for future work.

2 Unlearning Despite Mild Regularization

The results presented in Figure 1 and Table 1 show that UNCLE outperforms Guo et al. (2020), but they do not explain *why* UNCLE performs so well. We hypothesize that the performance of UNCLE stems from its ability to certify tight bounds on the approximation error of a Newton step, despite using only mild regularization. Here we support this hypothesis empirically.

Setup In Figure 1, we selected the hyper-parameters of both unlearning algorithms through a careful scan over the entire hyper-parameter space (see Appendix C). To measure the effect of increased regularization, we consider what happens when all algorithms (baseline training without unlearning, Guo et al. (2020), and UNCLE) are all forced to use a higher regularization coefficient.

In Figure 3, we fix a deletion request size $k = 100$, and vary a "regularization factor": All hyper-parameter searches are rerun for each such factor, but confined to regularization levels at least the "regularization factor" times the test-loss-minimizing regularization coefficient.

Results As can be seen from Figure 3, in most of the datasets UNCLE significantly outperforms (Guo et al., 2020) for baseline regularization, often nearly matching the test accuracy of the underlying learning problem. As a higher regularization is forced on the system, both the baseline learning and UNCLE lose much of their accuracy to the increased bias, but Guo et al. (2020) is unchanged until the restriction on regularization begins excluding its preferred values.

This supports the hypothesis that much of the advantage of UNCLE over Guo et al. (2020) can be attributed to UNCLE's tolerance of the mild regularization coefficients typically chosen in practice.

3 Discussion and Limitations

Summary of Our Contributions Our work revisits a central bottleneck in machine unlearning: even when a fast and accurate heuristic for approximate unlearning is available, truly masking the samples being removed requires the addition of well-calibrated noise. Prior methods are either heuristic or work only for learning problems where provable bounds on the error of such an approximation follow from global strong-convexity – a property which are not satisfied even by logistic regression.

We address these issues by replacing global, worst-case bounds on the error of approximate unlearning methods with local and geometry-aware ones in the setting of logistic regressions. We demonstrate empirically that this leads to an unlearning algorithm, UNCLE, which can unlearn many more samples than baseline methods with a smaller impact test accuracy.

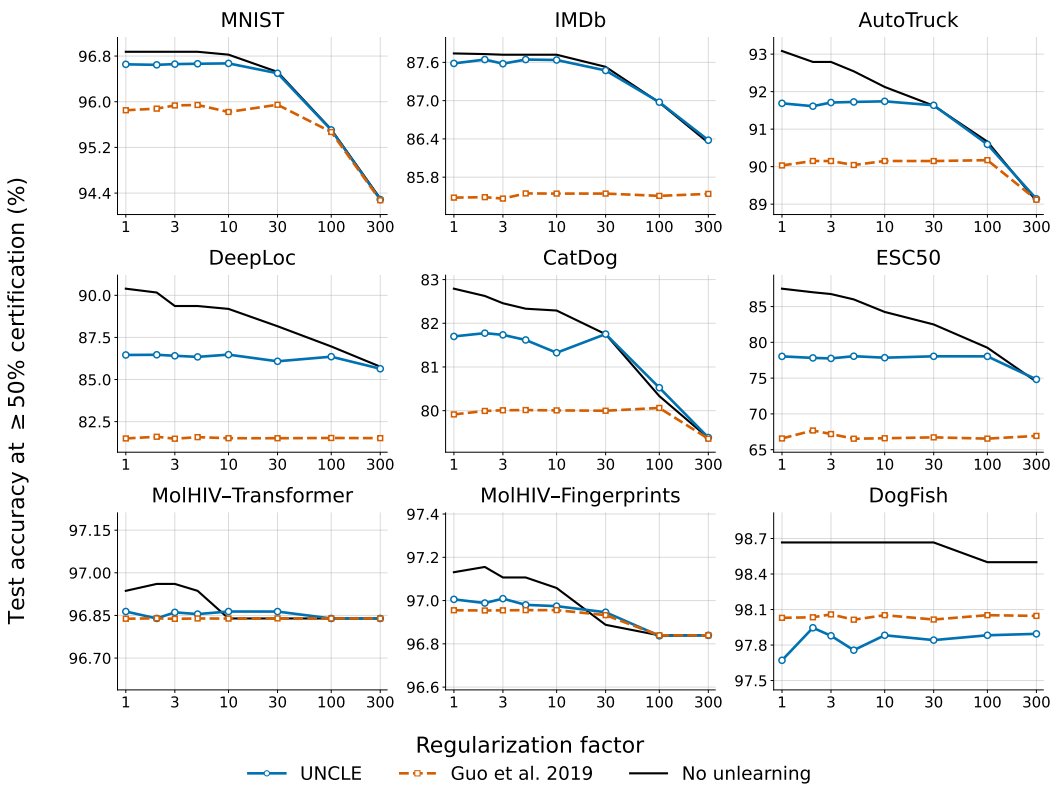


Figure 3: Fixing unlearning set size $k = 100$, privacy parameters $\varepsilon = 1, \delta = 10^{-4}$ and a retrain rate budget of $\leq 50\%$, we vary a "regularization factor": a lower limit on the range of allowed regularization coefficients. For each regularization factor and unlearning algorithm, we do a hyper-parameter scan and plot the highest achieved test accuracy with a retrain rate of at most 50%. For most datasets, UNCLE outperforms Guo et al. (2020) for the native regularization (regularization factor 1), often nearly matching that of no unlearning, but as the lower limit is increased, the test accuracy of UNCLE converges to (Guo et al., 2020). This supports the interpretation that much of the advantage of UNCLE lies in its ability to operate with mild regularization coefficients.

Future Work We focus here on the simpler setting of unlearning for logistic regression, where we can leverage a theoretical analysis to derive masking operations that are guaranteed to hide the unlearned samples with a small impact to the model's accuracy. A natural next step is to transfer these algorithmic improvements to the neural net setting. A second important avenue for future work is to use the innovations of `FoundCertificate` to improve *differentially private* algorithms for logistic regression – `FoundCertificate` offers an avenue to avoid DP-SGD by adding privacy-preserving noise to the output of a logistic regression directly proportional to its sensitivity to sample removals, similar to the approach of Rubinstein et al. (2026) for private linear regression.

Limitations While we hope our insights can inspire future algorithms for unlearning for non-convex models, this paper only directly addresses certified unlearning for logistic regression. Our experiments compare only to the baseline method of Guo et al. (2020), largely because this was the only suitable method we found in the literature (see Appendix A). The no-retrain query time of UNCLE is slower than that of Guo et al. (2020) (see Table 1), though in practice both are dominated by need to call a full retrain when certification fails. Part of this difference in query times is due to the fact that UNCLE must perform a pass over the retained samples in its certification step (by contrast, while Guo et al. (2020) must also retain the input dataset, they only access it to perform full retrains when certification fails). Finally, while we prove correctness of our algorithm and evaluate its performance empirically, we leave it to future work to provide an asymptotic analysis of its performance.

4 AI Disclosure and Reproducibility Statement

AI Disclosure In this work, we used generative AI tools to formulate mathematical claims, to assist in the writing of proofs, and to implement methods. We have not used generative AI tools for developing theoretical models or conceptual frameworks, providing critical ingredients for proving mathematical claims, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, assisting with translation, cleaning or reformatting datasets, supporting qualitative and thematic data analysis, or interpreting results. Additionally, we used generative AI tools for creating figures[4], creating software code, drafting proofs and several paragraphs in our paper, and identifying relevant literature. We have reviewed and substantially edited all AI-assisted work, rewriting proof drafts for human readability and rewriting all AI-drafted prose. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

Reproducibility Statement Full details of the UNCLE algorithm are available in Appendix B. We provide detailed mathematical proofs of the theorems in our paper in Appendices B and D. We provide a detailed overview of our experimental design, including a list of all datasets, pre-processing methods, and hyper-parameter selections in Appendix C.

References

José Juan Almagro Armenteros, Casper Kaae Sønderby, Søren Kaae Sønderby, Henrik Nielsen, and Ole Winther. DeepLoc: prediction of protein subcellular localization using deep learning. *Bioinformatics*, 33(21):3387–3395, 2017. doi: 10.1093/bioinformatics/btx431. URL https://doi.org/10.1093/bioinformatics/btx431.

Prashanti Anderson, Ainesh Bakshi, Mahbod Majid, and Stefan Tiegel. Sample-optimal private regression in polynomial time. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, pp. 2341–2349. Association for Computing Machinery, 2025. doi: 10.1145/3717823.3718218. URL https://doi.org/10.1145/3717823.3718218.

Martin Bertran, Shuai Tang, Michael Kearns, Jamie Morgenstern, Aaron Roth, and Zhiwei Steven Wu. Reconstruction attacks on machine unlearning: Simple models are vulnerable. In *Advances in Neural Information Processing Systems*, volume 37, pp. 104995–105016. Curran Associates, Inc., 2024. doi: 10.52202/079017-3334. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/bd996108ed57d388866ca6deb7acf6cb-Abstract-Conference.html.

Vijay Bhattiprolu, Mrinalkanti Ghosh, Venkatesan Guruswami, Euiwoong Lee, and Madhur Tulsiani. Weak decoupling, polynomial folds and approximate optimization over the sphere. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 1008–1019. IEEE, 2017. doi: 10.1109/FOCS.2017.97. URL https://doi.org/10.1109/FOCS.2017.97.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, Proceedings - IEEE Symposium on Security and Privacy, pp. 141–159. IEEE, 2021. doi: 10.1109/SP40001.2021.00019. URL https://doi.org/10.1109/SP40001.2021.00019.

Gavin Brown, Samuel B. Hopkins, and Adam Smith. Fast, sample-efficient, affine-invariant private mean and covariance estimation for subgaussian distributions. In Gergely Neu and Lorenzo Rosasco (eds.), *Proceedings of Thirty Sixth Conference on Learning Theory*, volume 195 of *Proceedings of Machine Learning Research*, pp. 5578–5579. PMLR, 2023. URL https://proceedings.mlr.press/v195/brown23a.html.

Gavin Brown, Jonathan Hayase, Samuel Hopkins, Weihao Kong, Xiyang Liu, Sewoong Oh, Juan C. Perdomo, and Adam Smith. Insufficient statistics perturbation: Stable estimators for private least

[4]These figures were generated using Python / TikZ code and AI tools were used to write this code, but no figures were generated directly by AI.

squares extended abstract. In Shipra Agrawal and Aaron Roth (eds.), *Proceedings of Thirty Seventh Conference on Learning Theory*, volume 247 of *Proceedings of Machine Learning Research*, pp. 750–751. PMLR, 2024. URL https://proceedings.mlr.press/v247/brown24b.html.

Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE Symposium on Security and Privacy*, SP '15, pp. 463–480. IEEE Computer Society, 2015. doi: 10.1109/SP.2015.35. URL https://doi.org/10.1109/SP.2015.35.

D. F. Davidenko. On a new method of numerical solution of systems of nonlinear equations. *Doklady Akademii Nauk SSSR*, 88(4):601–602, 1953. In Russian.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. *Journal of Privacy and Confidentiality*, 7(3):17–51, 2017. doi: 10.29012/jpc.v7i3.405. URL https://journalprivacyconfidentiality.org/index.php/jpc/article/view/405.

Jack Foster, Stefan Schoepf, and Alexandra Brintrup. Fast machine unlearning without retraining through selective synaptic dampening. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 12043–12051, 2024.

Antonio A. 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*, NeurIPS '19, pp. 3518–3531. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/hash/cb79f8fa58b91d3af6c9c991f63962d3-Abstract.html.

Ryan Giordano, William Stephenson, Runjing Liu, Michael I. Jordan, and Tamara Broderick. A Swiss Army infinitesimal jackknife. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pp. 1139–1147. PMLR, 2019. URL https://proceedings.mlr.press/v89/giordano19a.html.

Shashwat Goel, Ameya Prabhu, Amartya Sanyal, Ser-Nam Lim, Philip Torr, and Ponnurangam Kumaraguru. Towards adversarial evaluations for inexact machine unlearning. *arXiv preprint arXiv:2201.06640*, 2022.

Laura Graves, Vineel Nagisetty, and Vijay Ganesh. Amnesiac machine learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 11516–11524, 2021.

Sam Gunn. How to sketch a learning algorithm, 2026. URL https://arxiv.org/abs/2604.07328.

Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens van der Maaten. Certified data removal from machine learning models. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3832–3842. PMLR, 2020. URL https://proceedings.mlr.press/v119/guo20c.html.

Christopher J. Hillar and Lek-Heng Lim. Most tensor problems are NP-hard. *Journal of the ACM*, 60(6):45:1–45:39, 2013. doi: 10.1145/2512329. URL https://doi.org/10.1145/2512329.

Samuel B. Hopkins, Jonathan Shi, and David Steurer. Tensor principal component analysis via sum-of-square proofs. In Peter Grünwald, Elad Hazan, and Satyen Kale (eds.), *Proceedings of The 28th Conference on Learning Theory*, volume 40 of *Proceedings of Machine Learning Research*, pp. 956–1006, Paris, France, 2015. PMLR. URL https://proceedings.mlr.press/v40/Hopkins15.html.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open Graph Benchmark: Datasets for machine learning on graphs. In *Advances in Neural Information Processing Systems*, volume 33, pp. 22118–22133. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/fb60d411a5c5b72b2e7d3527cfc84fd0-Abstract.html.

Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. Knowledge unlearning for mitigating privacy risks in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14389–14408, 2023.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1885–1894. PMLR, 2017. URL https://proceedings.mlr.press/v70/koh17a.html.

Pang Wei Koh, Kai-Siang Ang, Hubert H. K. Teo, and Percy Liang. On the accuracy of influence functions for measuring group effects. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 5255–5265, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/a78482ce76496fcf49085f2190e675b4-Abstract.html.

Anastasia Koloskova, Youssef Allouah, Animesh Jha, Rachid Guerraoui, and Sanmi Koyejo. Certified unlearning for neural networks. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 31275–31298. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/koloskova25a.html.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. Towards unbounded machine unlearning. *Advances in neural information processing systems*, 36:1957–1987, 2023.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791. URL https://doi.org/10.1109/5.726791.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150. Association for Computational Linguistics, 2011. URL https://aclanthology.org/P11-1015/.

Ananth Mahadevan and Michael Mathioudakis. Certifiable unlearning pipelines for logistic regression: An experimental study. *Machine Learning and Knowledge Extraction*, 4(3):591–620, 2022. doi: 10.3390/make4030028. URL https://www.mdpi.com/2504-4990/4/3/28.

Seth Neel, Aaron Roth, and Saeed Sharifi-Malvajerdi. Descent-to-delete: Gradient-based methods for machine unlearning. In Vitaly Feldman, Katrina Ligett, and Sivan Sabato (eds.), *Proceedings of the 32nd International Conference on Algorithmic Learning Theory*, volume 132 of *Proceedings of Machine Learning Research*, pp. 931–962. PMLR, 2021. URL https://proceedings.mlr.press/v132/neel21a.html.

Aaradhya Pandey, Arnab Auddy, Haolin Zou, Arian Maleki, and Sanjeev Kulkarni. Gaussian certified unlearning in high dimensions: A hypothesis testing approach. In *International Conference on Learning Representations*, volume 2026, pp. 123230–123267, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/c83d21e25c0ef648d2cfbed8a63ceaa0-Paper-Conference.pdf.

Martin Pawelczyk, Jimmy Z. Di, Yiwei Lu, Gautam Kamath, Ayush Sekhari, and Seth Neel. Machine unlearning fails to remove data poisoning attacks. In *Proceedings of the 13th International Conference on Learning Representations*, ICLR '25, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/7e810b2c75d69be186cadd2fe3febeab-Abstract-Conference.html.

Karol Jerzy Piczak. ESC: Dataset for environmental sound classification. In *Proceedings of the 23rd ACM International Conference on Multimedia*, pp. 1015–1018. ACM, 2015. doi: 10.1145/2733373.2806390. URL https://dl.acm.org/doi/10.1145/2733373.2806390.

Kamiar Rahnama Rad and Arian Maleki. A scalable estimate of the out-of-sample prediction error via approximate leave-one-out cross-validation. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 82(4):965–996, 2020. doi: 10.1111/rssb.12374. URL https://academic.oup.com/jrsssb/article/82/4/965/7056054.

Ittai Rubinstein and Samuel B. Hopkins. Rescaled influence functions: Accurate data attribution in high dimension. In *Advances in Neural Information Processing Systems*, volume 38, pp. 150915–150944. Curran Associates, Inc., 2025. doi: 10.52202/085713-5047. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/ddbbcd937d63d5c6b935c07b1a8222ec-Abstract-Conference.html.

Ittai Rubinstein and Samuel B. Hopkins. On the accuracy of Newton step and influence function data attributions. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*. PMLR, 2026. URL https://arxiv.org/abs/2512.12572.

Ittai Rubinstein, Chris Ge, and Samuel B. Hopkins. Private linear regression via a down-sensitivity to privacy reduction. In *Proceedings of Thirty Ninth Conference on Learning Theory*, volume 336 of *Proceedings of Machine Learning Research*, pp. 5662–5720. PMLR, 2026. URL https://proceedings.mlr.press/v336/rubinstein26a.html.

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*, volume 34, pp. 18075–18086, 2021. URL https://papers.nips.cc/paper/2021/hash/9627c45df543c816a3ddf2d8ea686a99-Abstract.html.

Ashia Wilson, Maximilian Kasy, and Lester Mackey. Approximate cross-validation: Guarantees for model assessment and selection. In Silvia Chiappa and Roberto Calandra (eds.), *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 4530–4540. PMLR, 2020. URL https://proceedings.mlr.press/v108/wilson20a.html.

Haolin Zou, Arnab Auddy, Yongchan Kwon, Kamiar Rahnama Rad, and Arian Maleki. Certified machine unlearning under high dimensional regime. *Journal of Machine Learning Research*, 26 (308):1–58, 2025. URL https://www.jmlr.org/papers/v26/25-1751.html.

A Additional Unlearning Algorithms

In building our benchmark, our goal was to select certified unlearning algorithms that had a clear unlearning-accuracy tradeoff (without needing to artificially add a runtime limit), and which could be readily implemented based on their paper. We used Guo et al. (2020), which we believe to be the previous state of the art for this setting as our experimental baseline. We did not compare to (Mahadevan & Mathioudakis, 2022; Gunn, 2026; Pandey et al., 2026) as they use a different notion of certified unlearning, so would not have been an apples-to-apples comparison. Below we list the reasons that we decided to leave comparisons with (Sekhari et al., 2021; Neel et al., 2021; Koloskova et al., 2025) for future work.

Sekhari et al. (2021) Sekhari et al. (2021) introduce a general framework for unlearning, but as we tried to convert their algorithms into concrete implementations, we encountered two main difficulties.

First, Sekhari et al. (2021) require adding noise whose scale depends on the Lipschitz coefficient of the Hessian, which in turn requires bounding a third-order tensor's operator norm (see Appendix E). Moreover, Sekhari et al. (2021) require the loss to be both strongly convex and Lipschitz in a neighborhood $\mathcal{W}$ that should contain the full retrain, but do not detail any procedure for selecting $\mathcal{W}$ (setting $\mathcal{W} = \mathbb{R}^d$ would produce null results, as no function can be both globally strongly convex and globally Lipschitz).

Therefore, any future benchmark that wishes to include (Sekhari et al., 2021) will need to answer these questions: *How should one bound the Lipschitzness of the Hessian?* and *How should one select the region $\mathcal{W}$?*

Neel et al. (2021) The unlearning method of Neel et al. (2021) is based on generating an approximate unlearned model by taking a variable number of gradient descent steps and then adding noise corresponding to a bound on the approximation error. Taking too few gradient steps would cause their algorithm to add too much noise, unreasonably biasing our comparison against them. Taking too many steps would allow them to unlearn with very little noise, since from a certain point this will converge to a full retrain, but this comes at a cost to the overall runtime. Therefore, any future benchmark that includes a comparison to Neel et al. (2021) would also need to answer the question: *how should we select the budget for an unlearning algorithm that offers a continuous tradeoff between efficient unlearning and a full retrain?*

Koloskova et al. (2025) Koloskova et al. (2025) propose a certified unlearning algorithm that uses noisy fine-tuning to update the model for a set of data deletions. Their method has the advantage that it allows them to certifiably unlearn a set of points, even if the loss function is entirely non-convex, allowing them to perform certified unlearning even for neural nets. They pay for this advantage with an increased runtime compared to approximate-and-noise methods – in their experiments, they report a $20\% - 50\%$ reduction in runtime relative to retraining from scratch, whereas queries to Guo et al. (2020) and UNCLE certifications were $> 10\times$ faster than a full retrain on almost all of the datasets in our study (see Table 1). Therefore, any future benchmark that wishes to compare with this study will need to answer the same fundamental question as for (Neel et al., 2021): *what is the cutoff for runtimes allowed for an unlearning algorithm in the study?*

B The UNCLE Algorithm and Proof of Theorem 1.2

The key component of the UNCLE algorithm (Algorithm 2) not yet defined is its certification finding step, `FoundCertificate`. The goal of this step is to verify that the conditions for applying Theorem 1.1 hold with respect to a sufficiently small set $\mathcal{A}$. In this section, we will describe the algorithm, analyze its complexity and prove that, true to its name, it certifies an upper bound on the distance between $\boldsymbol{\theta}_0$ and the optimum of the loss and bound its runtime (see Theorem B.1).

Theorem B.1 (UNCLE certification correctness)**.** `FoundCertificate` *(Algorithm 3) is sound – i.e., if* `FoundCertificate` *accepts, then*

$$\left\| \mathbf{Q}_\tau^{1/2} \left(\hat{\boldsymbol{\theta}} - \operatorname*{argmin}_{\boldsymbol{\theta} \in \mathbb{R}^d} \{L_R(\boldsymbol{\theta})\} \right) \right\|_2 \leq R_2 \,.$$

In Section B.1, we will show that Theorem B.1 implies that UNCLE indeed performs certified unlearning. In Section B.2, we will detail the list of checks that make up the certification step. In Section B.3, we will prove the correctness of the certification algorithm.

B.1 Theorem B.1 implies Theorem 1.2

In this section, we will show that the soundness of the certification algorithm (Theorem B.1) implies UNCLE is a certified unlearning algorithm (Theorem 1.2). In particular, we will show that:

Theorem B.2 (UNCLE certified unlearning)**.** *Fix the algorithm's hyperparameters, with $R_2, \tau > 0$, $1 \geq \varepsilon > 0$, and $0 < \delta < 1$. For any dataset S, deletion set $T \subseteq S$, and initial model $\boldsymbol{\theta}_0 \in \mathbb{R}^d$, write $R = S \setminus T$ and*

$$\boldsymbol{\theta}^\star \coloneqq \operatorname*{argmin}_{\boldsymbol{\theta} \in \mathbb{R}^d} \{L_R(\boldsymbol{\theta})\} \,, \qquad \sigma^2 \coloneqq 2 \frac{\log(1.25/\delta)}{\varepsilon^2} R_2^2 \,.$$

Then the output $\boldsymbol{\theta}'$ of UNCLE *(Algorithm 2) satisfies*

$$\boldsymbol{\theta}' \approx_{\varepsilon,\delta} \boldsymbol{\theta}^\star + \boldsymbol{\nu}^\star \,, \qquad \boldsymbol{\nu}^\star \sim \mathcal{N}\left(\mathbf{0}, \sigma^2 \mathbf{Q}_\tau^{-1}\right) \,,$$

in the sense of Definition 1.

Proof of Theorem B.2. Let $\hat{\boldsymbol{\theta}}$ denote the model immediately before adding noise. If `FoundCertificate` accepts, Theorem B.1 gives

$$\left\| \mathbf{Q}_\tau^{1/2} \left(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star \right) \right\|_2 \leq R_2 \,.$$

Otherwise, full retraining sets $\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}^\star$, so the same bound holds.

For fixed R, both output distributions use the same matrix $\mathbf{Q}_\tau \succ 0$. Multiplication by $\mathbf{Q}_\tau^{1/2}$ transforms their masking noise into $\mathcal{N}\left(\mathbf{0}, \sigma^2 \mathbf{I}_d\right)$, while their means are at Euclidean distance at most R_2. The standard Gaussian mechanism bound therefore gives

$$\mathbf{Q}_\tau^{1/2} \boldsymbol{\theta}' \approx_{\varepsilon,\delta} \mathbf{Q}_\tau^{1/2} \left(\boldsymbol{\theta}^\star + \boldsymbol{\nu}^\star \right) \,.$$

Applying the common inverse map $\mathbf{Q}_\tau^{-1/2}$ proves the claim. □

B.2 The Certification Algorithm

In this section, we describe `FoundCertificate` for logistic regression.

Notation Write $S = [n]$, $T \subseteq S$ be a deletion set, $R = S \setminus T$ the set of indices of retained samples, $\mathbf{X}_I = (\mathbf{x}_i^\intercal)_{i \in I}$ be the submatrix of samples corresponding to a subset of indices $I \subseteq S$, and $\mathbf{X} = \mathbf{X}_S$. For logistic loss $\ell(t, y) = \log(1 + e^t) - yt$, labels $Y_1, \ldots, Y_n$, and a regularization parameter $\lambda > 0$, let

$$L_I(\boldsymbol{\theta}) = \sum_{i \in I} \ell(\mathbf{x}_i^\intercal \boldsymbol{\theta}, Y_i) + \frac{\lambda}{2} \|\boldsymbol{\theta}\|_2^2, \qquad w(t) = \frac{e^t}{(1 + e^t)^2},$$
$$\mathbf{Q} = \mathbf{X}_R^\intercal \mathbf{X}_R + \tau \mathbf{I}, \qquad \mathbf{B} = \mathbf{X} [\nabla^2 L_S(\boldsymbol{\theta}_0)]^{-1} \mathbf{X}^\intercal.$$

Strategy `FoundCertificate` will ensure a bound on the distance between an initial hypothesis $\hat{\boldsymbol{\theta}}$ and the true minimizer $\boldsymbol{\theta}^\star$ by applying Theorem 1.1 in the special case where $\mathcal{A}$ (the neighborhood of $\hat{\boldsymbol{\theta}}$ where strong convexity of the loss is required) is given by

$$\mathcal{A} = \underbrace{\left\{ \boldsymbol{\theta} \mid \left\| \mathbf{Q}^{1/2} \left(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}} \right) \right\|_2 \leq R_2 \right\}}_{\text{the desired certificate}} \cap \underbrace{\left\{ \boldsymbol{\theta} \mid \left\| \mathbf{X}_R \left(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}} \right) \right\|_\infty \leq R_\infty \right\}}_{\text{allows for efficient verification}} .$$

By adding the L_∞ limitations to $\mathcal{A}$, we guarantee that the model's predictions do not change too much, which in turn allows us to bound the change to the Hessian of the loss in that neighborhood more efficiently.

Local Lipschitzness The logistic regression datasets we observed all had a relatively high test accuracy (see Table 1), which corresponds to a high certainty model (i.e., the observed predictions had large positive or negative logit values on most samples). In the large logit limit, the third-order derivative of CE loss with respect to the logits decay rapidly. Therefore, to get best results in practice[5], we will rely on "local Lipschitz coefficients":

$$m_{\text{global}} = \sup_{z \in \mathbb{R}} \{|w'(z)|\} \geq m_{\text{local}} = \max_{z \in I} \{|w'(z)|\} \,,$$

where $I \subseteq \mathbb{R}$ is a bounded interval.

Ideally, we would have wanted to assign each sample an interval of the form

$$\hat{I}_i \coloneqq [\hat{z}_i - R_\infty, \hat{z}_i + R_\infty] \,,$$

where $\hat{\mathbf{z}} = \mathbf{X}^\intercal \hat{\boldsymbol{\theta}}$, and using these intervals resulted in higher certification rates than observed with UNCLE. However, because $\hat{\boldsymbol{\theta}}$ depends on the deletion request, that would have pushed more of the computation from the dataset initialization (setup time) into the deletion dependent query time. Instead, we will use the intervals

$$I_i \coloneqq [z_i - s, z_i + s] \,,$$

where $\mathbf{z}$ represents the logits of the model before unlearning, and $s \geq R_\infty + \|\mathbf{z} - \hat{\mathbf{z}}\|_\infty$.

[5] In limited ablation experiments, even without this component UNCLE outperformed (Guo et al., 2020) on the MNIST, IMDB and CatDog datasets, but not on the DogFish dataset. The other datasets were not tested in this ablation.

The Certification Algorithm `FoundCertificate` takes as input a hyperparameter $s > 0$ and the auxiliary parameters defined below. The operator norm and diagonal entries of $\mathbf{B}$ depend only on the full dataset and initial (pre-unlearning) model and are precomputed in setup. For each fixed s, the bounds m_i and the operator norm and diagonal entries of $\mathbf{DBD}$, defined below, are also precomputed in setup. The bound q depends on the deletion set. We describe below how these bounds are computed and how s is selected. The auxiliary bounds are:

- Bounds on the derivatives of w in neighborhoods near $\mathbf{x}_i^\top \boldsymbol{\theta}_0$:

$$\mathbf{m} = (m_i)_{i \in S}, \qquad m_i \geq \sup_{|t - \mathbf{x}_i^\intercal \boldsymbol{\theta}_0| \leq s} |w'(t)|.$$

  In what follows, set $\mathbf{D} = \mathrm{Diag}((\sqrt{s m_i})_{i \in S})$ to be a diagonal matrix depending on the hyperparameter s and bounds m_i.

- A number q which upper-bounds the spectral norm of a matrix built out of the to-be-unlearned samples T:

$$0 \leq q < 1, \qquad q \geq \left\| \left(\sqrt{w(\mathbf{x}_i^\intercal \boldsymbol{\theta}_0) w(\mathbf{x}_j^\intercal \boldsymbol{\theta}_0)}\, (\mathbf{B})_{ij} \right)_{i,j \in T} \right\|_{\mathrm{op}}.$$

If we fail to certify that $q < 1$, the entire certification fails. For the case of logistic regression, m_i can be computed in constant time given $s, \mathbf{x}_i^\intercal \boldsymbol{\theta}_0$ and q is computed in time $O(k^3)$. Both computations are included in the reported runtime benchmarks ($\mathbf{m}$ is computed at setup time and q in query time).

We are finally ready to describe `FoundCertificate`.

Algorithm 3: FOUNDCERTIFICATE: certification for logistic regression.

Input: Data $(\mathbf{x}_i, Y_i) \in \mathbb{R}^d \times \{0,1\}$ for $i \in [n]$; deletion set $T \subsetneq [n]$; initial model $\boldsymbol{\theta}_0$; candidate $\hat{\boldsymbol{\theta}}$; parameters $\lambda, R_2, s, \tau > 0$; $\mathbf{m}$ and q satisfying the conditions above.

Output: **true** certifies $\|\mathbf{Q}^{1/2}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star)\|_2 \leq R_2$ and $\|\mathbf{X}_R(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star)\|_\infty \leq s - \|\mathbf{X}_R(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}_0)\|_\infty$, where $\boldsymbol{\theta}^\star = \arg\min_{\boldsymbol{\theta}} L_R(\boldsymbol{\theta})$; **false** means no certificate was found.

$R_\infty \leftarrow s - \|\mathbf{X}_R(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}_0)\|_\infty$

$\mathbf{v} \leftarrow [\nabla^2 L_R(\boldsymbol{\theta}_0)]^{-1} \nabla L_R(\hat{\boldsymbol{\theta}})$

$G_{\mathrm{op}} \leftarrow \frac{\|\mathbf{DBD}\|_{\mathrm{op}}}{1-q}$; $\quad G_{\mathrm{diag}} \leftarrow \frac{\max_{i \in S} (\mathbf{DBD})_{ii}}{1-q}$

$\beta \leftarrow \|\mathbf{D}[R, R]\mathbf{X}_R \mathbf{v}\|_2$

$\Lambda \leftarrow \frac{1}{\sqrt{1-q}} \min \left\{ \sqrt{G_{\mathrm{diag}} \max_{i \in S}(\mathbf{B})_{ii}}\, \frac{\|\mathbf{X}_R(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}_0)\|_2 + R_2}{s},\ \sqrt{G_{\mathrm{op}} \|\mathbf{B}\|_{\mathrm{op}}} \right\}$

// Check the four inequalities in order.

if $G_{\mathrm{op}} < 1$

and $G_{\mathrm{diag}} < 1$

and $\|\mathbf{Q}^{1/2}\mathbf{v}\|_2 + \dfrac{\beta \sqrt{\left(\frac{\|\mathbf{B}\|_{\mathrm{op}}}{1-q} + \frac{\tau}{\lambda} \right) G_{\mathrm{op}}}}{1 - G_{\mathrm{op}}} < R_2$

and $\|\mathbf{X}_R \mathbf{v}\|_\infty + \dfrac{\Lambda \beta}{1 - G_{\mathrm{op}}} < (1 - G_{\mathrm{diag}}) R_\infty$ ***then***

  return true

return false

Theorem B.3 (Certification)**.** *Suppose the inputs to Algorithm 3 satisfy the conditions above, including the bounds on* $\mathbf{m}$ *and* q*. If* `FoundCertificate` *returns* ***true****, then the retained-data minimizer* $\boldsymbol{\theta}^\star$ *satisfies*

$$\|\mathbf{Q}^{1/2}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star)\|_2 \leq R_2, \qquad \|\mathbf{X}_R(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star)\|_\infty \leq s - \|\mathbf{X}_R(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}_0)\|_\infty.$$

B.3 Proof of Theorem B.3

Proof. Suppose that `FoundCertificate` returns **true**. Since $G_{\mathrm{op}}, G_{\mathrm{diag}} < 1$, the fourth check implies $R_\infty > 0$. We apply Theorem D.1 to L_R on the set

$$\mathcal{A} = \left\{ \boldsymbol{\theta} \in \mathbb{R}^d : \|\mathbf{Q}^{1/2}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})\|_2 < R_2, \quad \|\mathbf{X}_R(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})\|_\infty < R_\infty \right\}.$$

This is a convex neighborhood of $\hat{\boldsymbol{\theta}}$, and it is bounded because $\mathbf{Q} \succeq \tau \mathbf{I}$. Write $\mathbf{H}_{\boldsymbol{\theta}} = \nabla^2 L_R(\boldsymbol{\theta})$ and $\mathbf{H}_0 = \nabla^2 L_R(\boldsymbol{\theta}_0)$. The Hessian of L_R is Lipschitz on the compact closure of $\mathcal{A}$, and $\mathbf{H}_{\boldsymbol{\theta}} \succeq \lambda \mathbf{I} \succ (\lambda/2)\mathbf{I}$ everywhere. It remains to show that

$$\hat{\boldsymbol{\theta}} - \mathbf{H}_{\boldsymbol{\theta}}^{-1} \nabla L_R(\hat{\boldsymbol{\theta}}) \in \mathcal{A} \qquad \text{for every } \boldsymbol{\theta} \in \mathcal{A}. \tag{1}$$

Bounds on the Hessian. The matrix whose norm is bounded by q is the Gram matrix of the vectors $\sqrt{w(\mathbf{x}_i^\intercal \boldsymbol{\theta}_0)}[\nabla^2 L_S(\boldsymbol{\theta}_0)]^{-1/2}\mathbf{x}_i$, for $i \in T$. The sum of their outer products has the same nonzero eigenvalues, so the bound on q gives

$$\sum_{i \in T} w(\mathbf{x}_i^\intercal \boldsymbol{\theta}_0)\mathbf{x}_i \mathbf{x}_i^\intercal \preceq q \nabla^2 L_S(\boldsymbol{\theta}_0).$$

Subtracting this sum from the full-data Hessian yields

$$\mathbf{H}_0 \succeq (1-q)\nabla^2 L_S(\boldsymbol{\theta}_0), \qquad \mathbf{H}_0^{-1} \preceq \frac{[\nabla^2 L_S(\boldsymbol{\theta}_0)]^{-1}}{1-q}.$$

Thus the matrix

$$\mathbf{C} = \mathbf{X}_R \mathbf{H}_0^{-1} \mathbf{X}_R^\intercal$$

has rows and columns indexed by R and satisfies $0 \preceq \mathbf{C} \preceq \mathbf{B}[R,R]/(1-q)$. Taking operator norms and diagonal entries, and using the definitions of G_{op} and G_{diag}, gives

$$\begin{aligned} \|\mathbf{C}\|_{\mathrm{op}} &\le \frac{\|\mathbf{B}\|_{\mathrm{op}}}{1-q}, & \|\mathbf{D}[R,R]\mathbf{C}\mathbf{D}[R,R]\|_{\mathrm{op}} &\le G_{\mathrm{op}}, \\ \max_{i \in R} (\mathbf{C})_{ii} &\le \frac{\max_{i \in S}(\mathbf{B})_{ii}}{1-q}, & \max_{i \in R} (\mathbf{D})_{ii}^2 (\mathbf{C})_{ii} &\le G_{\mathrm{diag}}. \end{aligned} \tag{2}$$

Fix $\boldsymbol{\theta} \in \mathcal{A}$. Since $\mathbf{Q} \succeq \mathbf{X}_R^\intercal \mathbf{X}_R$, the triangle inequality gives

$$\begin{aligned} \|\mathbf{X}_R(\boldsymbol{\theta} - \boldsymbol{\theta}_0)\|_2 &\le \|\mathbf{X}_R(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}_0)\|_2 + R_2, \\ \|\mathbf{X}_R(\boldsymbol{\theta} - \boldsymbol{\theta}_0)\|_\infty &\le \|\mathbf{X}_R(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}_0)\|_\infty + R_\infty \le s. \end{aligned}$$

For each $i \in R$, the segment between $\mathbf{x}_i^\intercal \boldsymbol{\theta}_0$ and $\mathbf{x}_i^\intercal \boldsymbol{\theta}$ therefore lies in the interval on which m_i bounds $|w'|$. The mean value theorem implies

$$|w(\mathbf{x}_i^\intercal \boldsymbol{\theta}) - w(\mathbf{x}_i^\intercal \boldsymbol{\theta}_0)| \le m_i |\mathbf{x}_i^\intercal (\boldsymbol{\theta} - \boldsymbol{\theta}_0)|.$$

Let $\mathbf{F}$ be diagonal, with rows and columns indexed by R, and set $(\mathbf{F})_{ii} = [w(\mathbf{x}_i^\intercal \boldsymbol{\theta}) - w(\mathbf{x}_i^\intercal \boldsymbol{\theta}_0)]/(s m_i)$ when $m_i > 0$, and $(\mathbf{F})_{ii} = 0$ otherwise. The preceding inequality gives $w(\mathbf{x}_i^\intercal \boldsymbol{\theta}) - w(\mathbf{x}_i^\intercal \boldsymbol{\theta}_0) = s m_i (\mathbf{F})_{ii}$ in both cases. It follows that

$$\mathbf{H}_{\boldsymbol{\theta}} = \mathbf{H}_0 + \mathbf{X}_R^\intercal \mathbf{D}[R,R]\mathbf{F}\mathbf{D}[R,R]\mathbf{X}_R, \tag{3}$$

and

$$\max_{i \in R} |(\mathbf{F})_{ii}| \le 1, \qquad \left(\sum_{i \in R} (\mathbf{F})_{ii}^2 \right)^{1/2} \le \frac{\|\mathbf{X}_R(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}_0)\|_2 + R_2}{s}. \tag{4}$$

The ℓ_2 bound. Let $\mathbf{h} = \mathbf{H}_{\boldsymbol{\theta}}^{-1} \nabla L_R(\hat{\boldsymbol{\theta}})$. By the definition of $\mathbf{v}$, we have $\mathbf{H}_{\boldsymbol{\theta}} \mathbf{h} = \mathbf{H}_0 \mathbf{v}$, so equation (3) gives

$$\mathbf{h} = \mathbf{v} - \mathbf{H}_0^{-1} \mathbf{X}_R^\intercal \mathbf{D}[R,R]\mathbf{F}\mathbf{D}[R,R]\mathbf{X}_R \mathbf{h}. \tag{5}$$

Multiplying by $\mathbf{D}[R,R]\mathbf{X}_R$ and using equation (2), $\|\mathbf{F}\|_{\mathrm{op}} \le 1$, and the definition of β, we obtain

$$\|\mathbf{D}[R,R]\mathbf{X}_R \mathbf{h}\|_2 \le \beta + G_{\mathrm{op}} \|\mathbf{D}[R,R]\mathbf{X}_R \mathbf{h}\|_2.$$

The first check, $G_{\text{op}} < 1$, therefore implies

$$\|\mathbf{D}[R,R]\mathbf{X}_R\mathbf{h}\|_2 \le \frac{\beta}{1-G_{\text{op}}}. \tag{6}$$

To bound the correction in equation (5) after multiplying by $\mathbf{Q}^{1/2}$, we use $\mathbf{C}^2 \preceq \|\mathbf{C}\|_{\text{op}}\mathbf{C}$ and $\mathbf{H}_0^{-2} \preceq \lambda^{-1}\mathbf{H}_0^{-1}$. The definition of $\mathbf{Q}$ then gives

$$\begin{aligned}\|\mathbf{Q}^{1/2}\mathbf{H}_0^{-1}\mathbf{X}_R^\intercal\mathbf{D}[R,R]\|_{\text{op}}^2 &= \left\|\mathbf{D}[R,R]\left(\mathbf{C}^2+\tau\mathbf{X}_R\mathbf{H}_0^{-2}\mathbf{X}_R^\intercal\right)\mathbf{D}[R,R]\right\|_{\text{op}}\\ &\le \left(\|\mathbf{C}\|_{\text{op}}+\frac{\tau}{\lambda}\right)\|\mathbf{D}[R,R]\mathbf{C}\mathbf{D}[R,R]\|_{\text{op}}\\ &\le \left(\frac{\|\mathbf{B}\|_{\text{op}}}{1-q}+\frac{\tau}{\lambda}\right)G_{\text{op}}.\end{aligned}$$

Combining this with equations (5) and (6) proves

$$\|\mathbf{Q}^{1/2}\mathbf{h}\|_2 \le \|\mathbf{Q}^{1/2}\mathbf{v}\|_2 + \frac{\beta\sqrt{\left(\frac{\|\mathbf{B}\|_{\text{op}}}{1-q}+\frac{\tau}{\lambda}\right)G_{\text{op}}}}{1-G_{\text{op}}} < R_2,$$

where the strict inequality is the third check.

The ℓ_∞ bound. Multiplying equation (5) by $\mathbf{x}_i^\intercal$, for $i \in R$, gives

$$\mathbf{x}_i^\intercal\mathbf{h} = \mathbf{x}_i^\intercal\mathbf{v} - \sum_{j\in R}(\mathbf{C})_{ij}(\mathbf{D})_{jj}(\mathbf{F})_{jj}(\mathbf{D}[R,R]\mathbf{X}_R\mathbf{h})_j.$$

We bound the term with $j = i$ separately from the other terms. Its absolute value is at most

$$(\mathbf{C})_{ii}(\mathbf{D})_{ii}^2|(\mathbf{F})_{ii}|\,|\mathbf{x}_i^\intercal\mathbf{h}| \le G_{\text{diag}}\|\mathbf{X}_R\mathbf{h}\|_\infty.$$

For the terms with $j \ne i$, use $(\mathbf{C})_{ij}^2 \le (\mathbf{C})_{ii}(\mathbf{C})_{jj}$, since $\mathbf{C} \succeq 0$. Together with equations (2) and (4), this implies

$$\begin{aligned}\sum_{\substack{j\in R\\ j\ne i}}(\mathbf{C})_{ij}^2(\mathbf{D})_{jj}^2(\mathbf{F})_{jj}^2 &\le (\mathbf{C})_{ii}\sum_{j\in R}(\mathbf{D})_{jj}^2(\mathbf{C})_{jj}(\mathbf{F})_{jj}^2\\ &\le \frac{\max_{j\in S}(\mathbf{B})_{jj}}{1-q}\,G_{\text{diag}}\left(\frac{\|\mathbf{X}_R(\hat{\boldsymbol{\theta}}-\boldsymbol{\theta}_0)\|_2+R_2}{s}\right)^2.\end{aligned}$$

We also have

$$\begin{aligned}\sum_{\substack{j\in R\\ j\ne i}}(\mathbf{C})_{ij}^2(\mathbf{D})_{jj}^2(\mathbf{F})_{jj}^2 &\le \sum_{j\in R}(\mathbf{C})_{ij}^2(\mathbf{D})_{jj}^2 \le \|\mathbf{C}\mathbf{D}[R,R]\|_{\text{op}}^2\\ &\le \|\mathbf{C}\|_{\text{op}}\|\mathbf{D}[R,R]\mathbf{C}\mathbf{D}[R,R]\|_{\text{op}} \le \frac{\|\mathbf{B}\|_{\text{op}}}{1-q}G_{\text{op}},\end{aligned}$$

using $|(\mathbf{F})_{jj}| \le 1$ and $\mathbf{C}^2 \preceq \|\mathbf{C}\|_{\text{op}}\mathbf{C}$. The smaller of these two bounds is Λ^2. By Cauchy–Schwarz, the terms with $j \ne i$ therefore have total absolute value at most $\Lambda\|\mathbf{D}[R,R]\mathbf{X}_R\mathbf{h}\|_2$. Taking the maximum over $i \in R$ yields

$$\|\mathbf{X}_R\mathbf{h}\|_\infty \le \|\mathbf{X}_R\mathbf{v}\|_\infty + G_{\text{diag}}\|\mathbf{X}_R\mathbf{h}\|_\infty + \Lambda\|\mathbf{D}[R,R]\mathbf{X}_R\mathbf{h}\|_2.$$

The second check allows us to rearrange this inequality; equation (6) and the fourth check then give

$$\|\mathbf{X}_R\mathbf{h}\|_\infty \le \frac{\|\mathbf{X}_R\mathbf{v}\|_\infty + \Lambda\beta/(1-G_{\text{op}})}{1-G_{\text{diag}}} < R_\infty.$$

Thus $\hat{\boldsymbol{\theta}} - \mathbf{h} \in \mathcal{A}$, proving equation (1). Theorem D.1 therefore gives a local minimum of L_R in $\mathcal{A}$. Since L_R is strongly convex, this is its unique minimizer $\boldsymbol{\theta}^\star$. The definition of $\mathcal{A}$, together with $R_\infty = s - \|\mathbf{X}_R(\hat{\boldsymbol{\theta}}-\boldsymbol{\theta}_0)\|_\infty$, proves both claimed bounds. □

C EXPERIMENT DESIGN

We compare logistic-regression unlearning on nine fixed-feature tasks, using $\varepsilon = 1$ and $\delta = 10^{-4}$ throughout. Feature extraction (e.g., using last-layer embeddings of a pretrained model) is performed once; removal requests affect the logistic-regression training samples, not the training data of a pretrained feature extractor. For a removal set T, the regularized objective is

$$L_{\setminus T}(\boldsymbol{\theta}) = \sum_{i \notin T} \left[\log(1 + \exp(X_i^\top \boldsymbol{\theta})) - Y_i X_i^\top \boldsymbol{\theta}\right] + \frac{\lambda}{2}\|\boldsymbol{\theta}\|_2^2.$$

Reported cross-entropy (CE) is the average unregularized loss on the evaluation set.

C.1 DATASETS

For most datasets, we followed the reproduction materials of Rubinstein & Hopkins (2025) for the original tasks and retain fixed features without additional standardization, PCA, or feature selection. Unless stated otherwise, a constant-one feature provides a ridge-regularized intercept; reported dimensions include it. The two MolHIV tasks compare representations of the same molecules, not independent data modalities.

MNIST. We use digits 3 versus 8 from MNIST LeCun et al. (1998), with labels 0 and 1, respectively. Images are flattened to 784 coordinates and divided by 255, without a learned embedding or appended intercept. We retain the frozen 6,000-example training subset, verified against the official training images, and use all 1,984 official test images of these two digits. The test partition was checked against the original image files; no validation subset is held out from the final training matrix.

IMDb. We use the positive/negative movie-review task Maas et al. (2011) and the existing 50,000-review feature archive. Following the procedures of Rubinstein & Hopkins (2025), these are divided into 40,000 training samples and 10,000 test samples using a fixed-seed permutation. A fixed BERT model is then used to convert the text reviews into 768 dimensional embeddings and an all-ones feature is added as an intercept.

AutoTruck and CatDog. We pool the official CIFAR-10 (Krizhevsky, 2009) training and test images of automobiles and trucks (respectively cats and dogs). Following the procedures of Rubinstein & Hopkins (2025), a frozen ResNet-18 is used to embed these images into feature vectors with 513 coordinates (including 1 intercept feature). A fixed seed permutation produces an 80/20 test-train split with 9,600 training and 2,400 test examples.

DeepLoc. We use membrane-versus-soluble labels from the DeepLoc 1.0 FASTA Almagro Armenteros et al. (2017), excluding samples with an "unknown" label. We preserve the official test markers and assign training-only sequence-similarity components to train/validation, giving 5,864/1,049/1,749 examples. Frozen ESM-2 (`esm2_t30_150M_UR50D`) features average final-layer residue embeddings, excluding special tokens, giving $d = 641$ with an intercept.

ESC50. We use all 2,000 ESC50 audio clips Piczak (2015). Following the procedures of Rubinstein & Hopkins (2025), we label them according to an artificial-versus-natural task, we generate 512-dimensional OpenL3 embeddings and an intercept feature is added yielding $d = 513$. A fixed-seed 80/20 train-test split was used, yielding 1,600 training samples and 400 test samples.

MolHIV–Transformer. We use ogbg-molhiv HIV-inhibition labels Hu et al. (2020), preserving molecule IDs and the official scaffold split of 32,901/4,113/4,113 train/validation/test examples. We use frozen `MoLFormer-XL-both-10pct` embeddings in deterministic evaluation mode. Its pooled 768-dimensional output and an intercept give $d = 769$.

MolHIV–Fingerprints. We use the same molecules, labels, ordering, and scaffold split as MolHIV–Transformer, but instead of a transformer we use direct feature extraction using RDKit 2024.3.5 to compute 2,048-bit, radius-two binary Morgan fingerprints directly from the original OGB SMILES,

with chirality and count simulation disabled, and adding an intercept yields $d = 2,049$. Fingerprints are verified to be invariant to molecule ordering and batch composition.

Both MolHIV tasks preserve a substantial class imbalance: an always-negative model has 96.84% test accuracy.

DogFish. We use the ImageNet-derived dog-versus-fish task of Koh & Liang (2017), retaining the released 2,048-dimensional Inception-v3 bottleneck features. The original balanced split has 1,800 training examples and 600 test examples. We add neither an intercept nor feature normalization. All current feature and label arrays match a fresh download of the original benchmark archives exactly, but we did not perform an independent reconstruction of their historical raw-image/checkpoint pipeline.

C.2 Baseline regularization

Before running either unlearning method, we fit unperturbed reference models on a logarithmic grid of regularization coefficients, then refine the search near its optima. We retain two references: $\lambda^\star_{\mathrm{CE}}$ minimizes selection-set CE, whereas $\lambda^\star_{\mathrm{acc}}$ maximizes selection-set accuracy. Loss increases and accuracy reductions use their respective reference models. The native unlearning search starts at $\lambda_{\min} = \min\{\lambda^\star_{\mathrm{CE}}, \lambda^\star_{\mathrm{acc}}\}$ and explicitly includes both reference values. The reference models and their selections are then frozen.

The completed benchmarks have three search histories. CatDog, IMDb, and DogFish refine a previously selected coarse reference λ_0 with 1,000 log-spaced values in $[0.05\lambda_0, 20\lambda_0]$. The original coarse selection used an internal training/validation split. DogFish then adds 1,000 smaller values at the same log spacing because its CE optimum hit the lower boundary. Current MNIST, AutoTruck, and ESC50 instead use 20 coarse values in $[10^{-8}, 10^2]$, followed by 1,000 log-spaced values from $0.05\min\{\lambda_{0,\mathrm{CE}}, \lambda_{0,\mathrm{acc}}\}$ to $20\max\{\lambda_{0,\mathrm{CE}}, \lambda_{0,\mathrm{acc}}\}$. DeepLoc and both MolHIV representations use 20 coarse values in $[10^{-6}, 10^2]$ and 20 fine values spanning both coarse optima and their adjacent coarse neighbors. Fine fits share the coarse-CE model as a warm start.

For the original six tasks, the fine search selects on the fixed test split; for the three biology tasks, it selects on the separate validation split, without folding validation samples into training. The latter's test curves are evaluated post hoc from saved models, without changing selections. These differences are preserved rather than presenting the existing results as a uniform validation-only protocol. Failed coarse fits are recorded while other candidates continue; two MNIST and three ESC50 failures at the smallest regularizations were reviewed before refinement. Figure 4 shows every sampled regularization location, including extensions and failures.

C.3 Why we report Pareto curves

For fixed data and removal count, each hyperparameter setting gives a retrain rate and a predictive utility. A setting is dominated if another has no larger retrain rate and no worse utility, with one strict improvement. The nondominated settings form an empirical Pareto frontier. Fixing a retrain budget selects the best feasible utility; fixing a CE-increase budget selects the smallest feasible retrain rate. These are alternative summaries of the same tradeoff, illustrated in Figure 5.

C.4 Guo et al.: coarse grid and refinement

For Guo et al. (2020), we search over regression regularization λ and objective-perturbation scale σ. Our implementation draws $b \sim \mathcal{N}(0, \sigma^2 I)$ and adds $b^\top \boldsymbol{\theta}/n$ to the objective above. Thus σ specifies the sum-objective perturbation, not the standard deviation of additive noise on model weights. The first stage evaluates a Cartesian grid in $(\log\lambda, \log\sigma)$.

The noise grid is $\sigma \in \{0.001, 0.003, 0.01, 0.03, 0.1, 0.3, 1, 3, 10, 30, 100, 300\}$. For seven tasks, the regularization grid has 32 log-spaced values between $\lambda_{\min}$ and $10^6\lambda_{\min}$, plus exact reference insertion, giving 33 distinct values. CatDog retains its completed narrower grid: 15 log-spaced values through $10^3\lambda_{\min}$, giving 16 after reference insertion. DogFish preserves its original 16 values and adds 16 above the old maximum through $10^6\lambda_{\min}$; it is not uniformly spaced across the entire enlarged interval.

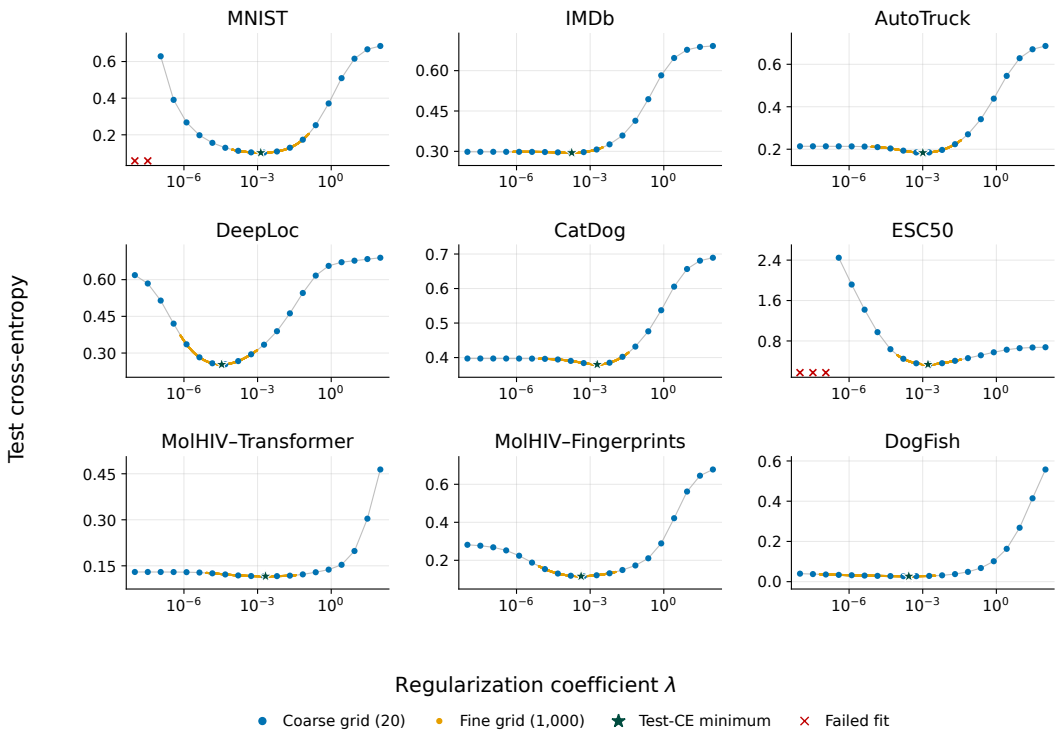


Figure 4: Baseline regularization searches. Each panel shows the evaluated regularization coefficients and the corresponding unperturbed test CE, distinguishing the initial search from refinement. Marked reference values were selected by their dataset's selection protocol; a test-CE minimum need not equal a validation-selected reference. For the historical CatDog, IMDb, and DogFish coarse scans, only validation losses survive; rug marks show their sampled regularizations without inventing test CE. Failed fits are likewise marked on the axis.

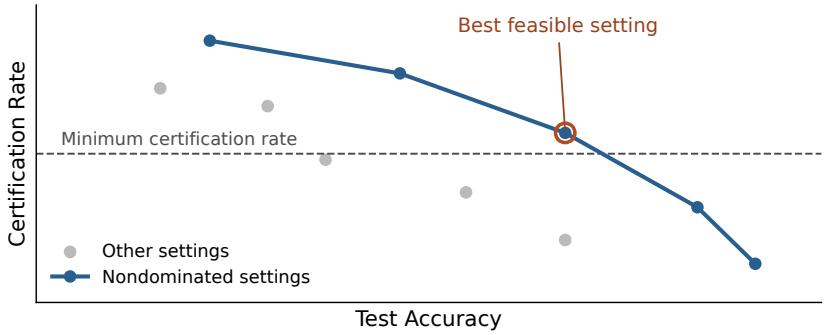


Figure 5: Illustrative Pareto tradeoff: upper-right is better. Each point is a hyperparameter setting; the highlighted point maximizes accuracy subject to the minimum certification rate. Lines guide the eye between evaluated settings.

Refinement concentrates computation near the coarse $k = 1$ CE frontier. We order its parameter tuples by CE cost, smooth the path in $(\log_{10}\lambda, \log_{10}\sigma)$ space with three endpoint-preserving corner-cutting steps, sample locations uniformly by path length, and add independent Gaussian offsets. Each offset's standard deviation is half the corresponding median coarse-grid log spacing. Out-of-range proposals are redrawn, not clipped. We evaluate the resulting tuples for every k and combine both stages. There are 1,188 refinement tuples for the seven larger grids and 576 each for CatDog and DogFish; DogFish's earlier refinement budget was preserved when its coarse range expanded. The smoothing changes proposal locations, not the empirical utility frontier.

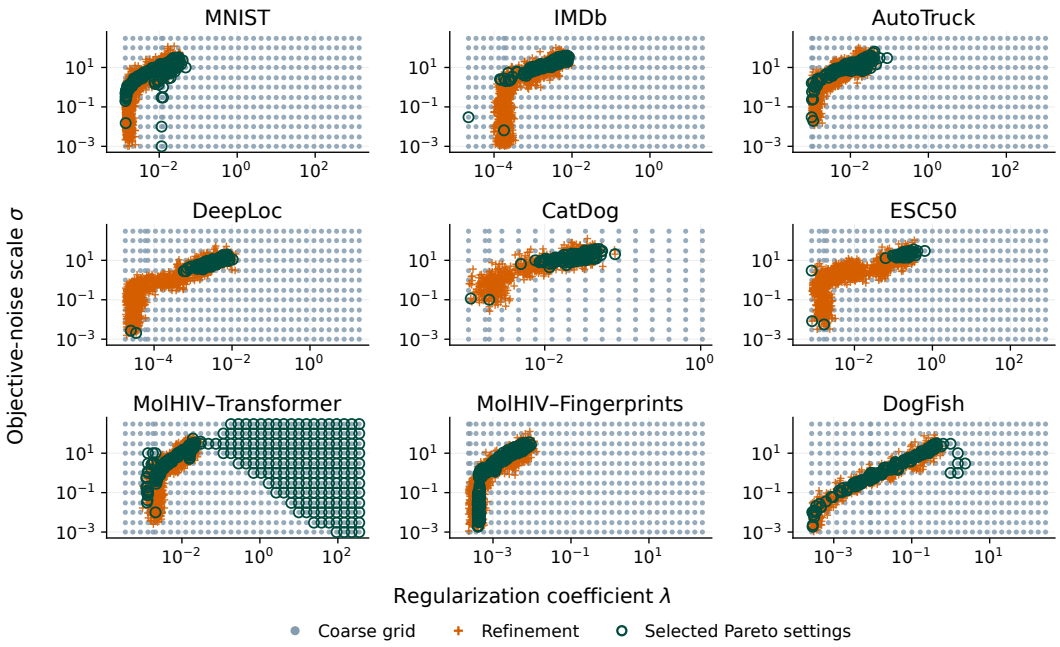


Figure 6: Guo et al. hyperparameter exploration. Coarse-grid and refinement candidates are shown in regression-regularization/objective-noise space. Highlighted settings lie on the discovery frontier for CE or accuracy at $k = 10$; they are hyperparameters, not points in utility/retrain-rate space. The refinement proposal itself is constructed from the coarse $k = 1$ CE frontier and is reused across removal counts.

C.5 UNCLE: HYPERPARAMETER SEARCH

We search over regression regularization, geometry ridge, and the allowed approximation radius. In particular,

$$\tau = cn\lambda, \qquad c \in \{10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}, \qquad R_2 = a\frac{kd}{n^{3/2}}, \quad a \in \mathrm{LogGrid}(10^{-2}, 10^{2}; 100),$$

where $\mathrm{LogGrid}(u, v; m)$ denotes m points equally spaced in logarithmic scale, including both endpoints. The normalization of R_2 sets a search range rather than an additional assumption in the certificate. For each setting, certification is an OR over the precommitted witness values $S_\infty \in \{10, 1, 10^{-1}, 10^{-2}, 10^{-3}\}$. Each τ uses its own matching certificate and masking geometry; we do not certify with one geometry and release with another.

For eight tasks, the regression grid contains ten log-spaced values in $[\lambda_{\min}, 10^3\lambda_{\min}]$, plus the other exact reference, giving eleven values. DogFish additionally has nine values extending to $10^6\lambda_{\min}$, giving twenty. Thus each k has $N_\lambda \times 5 \times 100$ operating points; the five S_∞ witnesses are checked within an operating point, not counted as separate noise settings. We do not apply Guo's frontier-guided parameter refinement to UNCLE.

Independent evaluation of selected settings. Both methods use 100 discovery seeds. Each seed supplies ten removal requests, with nested prefixes across k; certification uses all ten, while one predetermined request supplies the released-model utility for that seed. We freeze the union of each method's CE and accuracy frontier tuples for each k, then reevaluate them with 100 fresh seeds. We retain discovery membership and ordering rather than silently reselecting on fresh means. SEMs, where used, are over the 100 outer-seed estimates, not 1,000 independent utility outputs. This separates parameter discovery from repeat evaluation; it does not account for reference selection, fixed-test-set uncertainty, or the uncertainty of the entire optimized frontier. Unlearning frontiers still use test utility, even for validation-selected biology reference models.

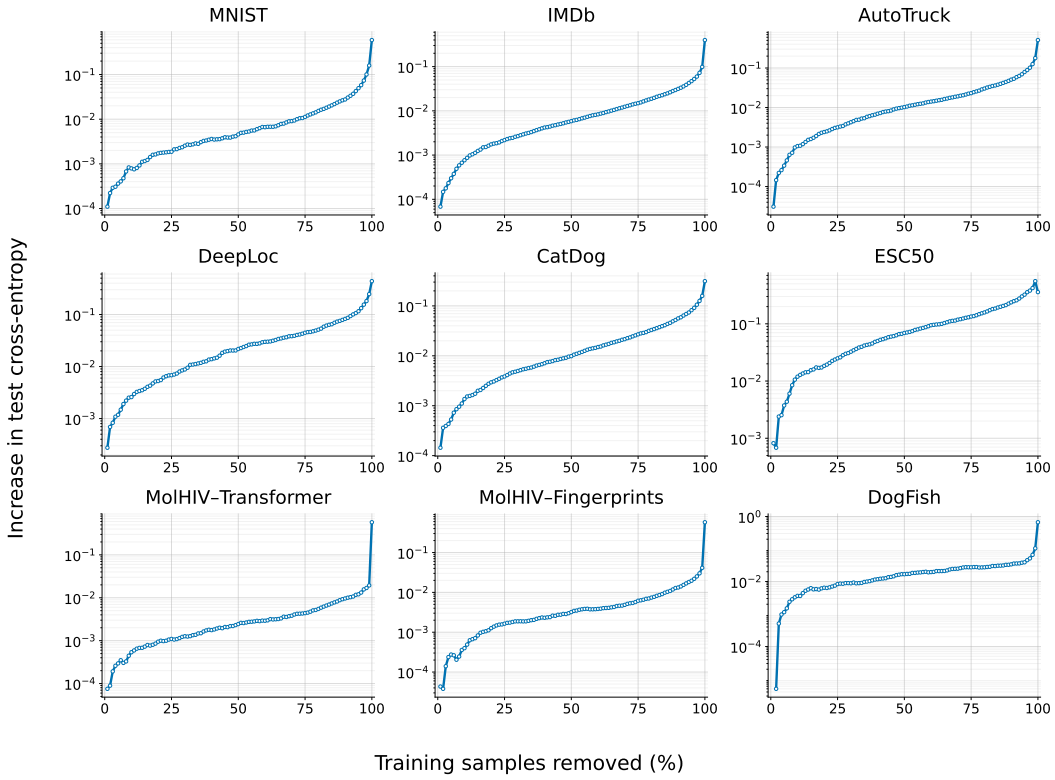


Figure 7: Random-removal calibration on all nine tasks. Curves show empirical mean test CE increase due to removal of a fraction of samples in each dataset. The empirical mean was taken over 30 random permutations and is reported as a function of the percentage of training samples removed.

C.6 Calibration by random data removal

To put utility degradation on a dataset-specific scale, we measure the effect of discarding training data and refitting without unlearning noise. For each dataset, we draw 30 independent random permutations and retain nested prefixes containing fractions $f \in \{0.01, 0.02, \ldots, 0.99\}$ of the training samples, rounding subset sizes to the nearest integer. We keep the test set and feature representation fixed.

For each subset, we fit logistic regression using the retained-sample mean loss and regularization coefficient λ_{cal}/f, where λ_{cal} is the full-data CE reference coefficient frozen when the calibration experiment was launched.

D Statement and Proof of Theorem 1.1 via Newton Flows

In this section we prove Theorem 1.1, the key tool underlying correctness of UNCLE which shows that local conditions of a loss function on any convex set can certify existence of a local minimum in that convex set.

The intuition is simple: if the gradient at $\hat{\boldsymbol{\theta}}$ is small (in the geometry imposed by the curvature of the loss), then we expect $\hat{\boldsymbol{\theta}}$ to be close to optimum. Conversely, if $\hat{\boldsymbol{\theta}}$ is close to the true optimum of some loss, then we would expect a step of Newton's algorithm to move it very little.

The challenge is to formalize these intuitions to obtain a concrete guarantee under assumptions that can be checked efficiently, and we do this with Theorem D.1, which states that as long as the gradient of the loss at $\hat{\boldsymbol{\theta}}$ is short in the geometry imposed by the loss curvature (evaluated at any point in a neighborhood of $\hat{\boldsymbol{\theta}}$), then $\hat{\boldsymbol{\theta}}$ is close to optimum of the loss:

Theorem D.1 (Formal version of Theorem 1.1)**.** *Let $L : \mathbb{R}^d \to \mathbb{R}$ be a smooth function, let $\hat{\boldsymbol{\theta}} \in \mathbb{R}^d$, and let $g \coloneqq \nabla L|_{\hat{\boldsymbol{\theta}}}$ be the gradient of L at $\hat{\boldsymbol{\theta}}$. (In our application to* UNCLE*, L is the loss function of a learning problem after removing some samples and $\hat{\boldsymbol{\theta}}$ is a candidate minimizer of L.)*

Suppose $\mathcal{A}$ is a bounded, convex neighborhood, $\hat{\boldsymbol{\theta}} \in \mathcal{A}$, and $M, \lambda > 0$ are such that every point $\boldsymbol{\theta} \in \mathcal{A}$ has an M-Lipschitz and positive-definite loss-curvature $\mathbf{H}_{\boldsymbol{\theta}} \coloneqq \nabla^2 L|_{\boldsymbol{\theta}} \succ \lambda \mathbf{I}$, and satisfies $\hat{\boldsymbol{\theta}} - \mathbf{H}_{\hat{\boldsymbol{\theta}}}^{-1} g \in \mathcal{A}$. Then $\mathcal{A}$ contains a local minimum of the loss L.

In particular, if the loss is also strongly convex everywhere, then this minimum is global and

$$\boldsymbol{\theta}^\star \coloneqq \operatorname*{argmin}_{\boldsymbol{\theta} \in \mathbb{R}^d} L(\boldsymbol{\theta}) \in \mathcal{A}\,.$$

To our knowledge, the closest precursor to Theorem D.1 in the machine learning literature is Rubinstein & Hopkins (2026), who prove that a similar bound holds whenever $\mathcal{A}$ is a ball centered around $\hat{\boldsymbol{\theta}}$. The fact that we can allow for a more general neighborhood $\mathcal{A}$ is not trivial. Indeed, the proof technique of Rubinstein & Hopkins (2026), which is based on tracking the gradient of the loss along a line from the initial guess $\hat{\boldsymbol{\theta}}$ to the edge of the neighborhood $\mathcal{A}$ and showing that it points outwards relies heavily on their assumption that $\mathcal{A}$ is a ball. Allowing for more general neighborhoods is necessary for our unlearning algorithm that will be based on an application of Theorem D.1 on the intersection of a ball with a convex polytope, requiring a different proof technique. More broadly, the technique of localizing a zero of a vector field (in this case the gradient field of the loss L) by integrating along the Newton flow dates at least to the 1950s (Davidenko, 1953).

Note that while Theorem D.1 assumes some degree of uniform strong convexity and Hessian-Lipschitzness within $\mathcal{A}$, the strong convexity and Lipschitz coefficients do not affect the quantitative guarantee of the theorem and are needed only for the sake of smoothness.

D.1 Proof sketch of Theorem D.1

Let $L, \hat{\boldsymbol{\theta}}, \mathbf{g}, \mathbf{H}$, and $\mathcal{A}$ be as in Theorem D.1. If the Hessian were constant in $\mathcal{A}$, then the Newton step

$$\hat{\boldsymbol{\theta}} - \mathbf{H}_{\hat{\boldsymbol{\theta}}}^{-1}\mathbf{g}$$

would yield a minimum of the loss. To allow for a Hessian that depends on the model parameters, we consider a continuous-time "Newton flow" – a function $\boldsymbol{\theta} : \mathbb{R}_{\geq 0} \to \mathbb{R}^d$ satisfying the differential equation

$$\frac{\mathrm{d}\boldsymbol{\theta}(t)}{\mathrm{d}t} = -\mathbf{H}_{\boldsymbol{\theta}(t)}^{-1} \nabla_{\boldsymbol{\theta}} L|_{\boldsymbol{\theta}(t)}, \qquad \boldsymbol{\theta}(0) = \hat{\boldsymbol{\theta}}\,.$$

To analyze this flow, it is useful to consider how the loss gradient changes as a function of t. Applying the chain rule, we have

$$\frac{\mathrm{d}}{\mathrm{d}t} \nabla_{\boldsymbol{\theta}} L|_{\boldsymbol{\theta}(t)} = \nabla_{\boldsymbol{\theta}} \left(\nabla_{\boldsymbol{\theta}} L\right)|_{\boldsymbol{\theta}(t)} \frac{\mathrm{d}\boldsymbol{\theta}(t)}{\mathrm{d}t} = -\mathbf{H}_{\boldsymbol{\theta}(t)} \mathbf{H}_{\boldsymbol{\theta}(t)}^{-1} \nabla_{\boldsymbol{\theta}} L|_{\boldsymbol{\theta}(t)} = -\nabla_{\boldsymbol{\theta}} L|_{\boldsymbol{\theta}(t)}\,.$$

In particular, this implies a clear and rapidly decaying structure of the gradient as a function of time $\nabla_{\boldsymbol{\theta}} L|_{\boldsymbol{\theta}(t)} = e^{-t} \nabla_{\boldsymbol{\theta}} L|_{\boldsymbol{\theta}(0)} \to 0$, meaning that it converges to a saddle-point of the loss. Moreover, from the Fundamental Theorem of Calculus (FTC), we have

$$\boldsymbol{\theta}(t) = \boldsymbol{\theta}(0) + \int_0^t \frac{\mathrm{d}\boldsymbol{\theta}(\tau)}{\mathrm{d}\tau} \mathrm{d}\tau = \boldsymbol{\theta}(0) - \int_0^t e^{-\tau} \mathbf{H}_{\boldsymbol{\theta}(\tau)}^{-1} \mathbf{g} \mathrm{d}\tau\,.$$

Our proof strategy will use this integral representation of $\boldsymbol{\theta}(t)$ together with the assumption that $\hat{\boldsymbol{\theta}} - \mathbf{H}_{\hat{\boldsymbol{\theta}}}^{-1}\mathbf{g} \in \mathcal{A}$ for all $\boldsymbol{\theta} \in \mathcal{A}$, to show that the Newton flow stays in $\mathcal{A}$, implying that the convergence point is inside $\mathcal{A}$ and also a minimum (and not just a saddle-point).

Formalizing this proof requires additional care (e.g., writing the Newton flow ODE does not automatically imply that it has a unique solution); we now proceed with this formalization.

D.2 Proof of Theorem D.1

We prove Theorem D.1 by analyzing the Newton flow introduced in Section D.1. We show that this flow is well defined for all times, stays within $\mathcal{A}$, and converges to a point in $\mathcal{A}$ with gradient $\mathbf{0}$. We state and prove three lemmas and then Theorem D.1 itself.

Lemma D.2 (The Newton Flow is Well-Defined and Stays in $\mathcal{A}$)**.** *Under the conditions of Theorem D.1, the initial value problem*

$$\boldsymbol{\theta}(0) = \hat{\boldsymbol{\theta}}, \qquad \dot{\boldsymbol{\theta}}(t) = -\mathbf{H}_{\boldsymbol{\theta}(t)}^{-1} \nabla L|_{\boldsymbol{\theta}(t)} \tag{7}$$

has a unique solution for all $t \geq 0$. Moreover,

$$\boldsymbol{\theta}(t) \in \operatorname{int}(\mathcal{A}) \qquad \textit{for every } t \geq 0.$$

Proof. The Hessian is Lipschitz and uniformly invertible on $\mathcal{A}$, so its inverse is locally Lipschitz there. Consequently, the vector field $\boldsymbol{\theta} \mapsto -\mathbf{H}_{\boldsymbol{\theta}}^{-1} \nabla L|_{\boldsymbol{\theta}}$ is locally Lipschitz on $\operatorname{int}(\mathcal{A})$. Since $\mathcal{A}$ is a neighborhood of $\hat{\boldsymbol{\theta}}$, we have $\hat{\boldsymbol{\theta}} \in \operatorname{int}(\mathcal{A})$. The Picard–Lindelöf Theorem therefore gives local existence and uniqueness. Let $[0, T)$, with $T \in (0, \infty]$, be the maximal interval on which this solution exists, remains unique, and retains values in $\operatorname{int}(\mathcal{A})$. Our goal is to prove $T = \infty$. Assume for the sake of contradiction that this is not the case.

For every $t < T$, the chain rule gives

$$\frac{\mathrm{d}}{\mathrm{d}t} \nabla L|_{\boldsymbol{\theta}(t)} = \mathbf{H}_{\boldsymbol{\theta}(t)} \dot{\boldsymbol{\theta}}(t) = -\nabla L|_{\boldsymbol{\theta}(t)} \,,$$

where we used the definition of $\dot{\boldsymbol{\theta}}(t)$ and canceled using $\mathbf{H}_{\boldsymbol{\theta}(t)} \cdot \mathbf{H}_{\boldsymbol{\theta}(t)}^{-1} = I$. Since $\mathbf{g} = \nabla L|_{\hat{\boldsymbol{\theta}}}$, we get the following exponential decays:

$$\nabla L|_{\boldsymbol{\theta}(t)} = e^{-t} \mathbf{g}, \qquad \dot{\boldsymbol{\theta}}(t) = -e^{-t} \mathbf{H}_{\boldsymbol{\theta}(t)}^{-1} \mathbf{g}. \tag{8}$$

The Hessian lower bound therefore implies

$$\left\| \dot{\boldsymbol{\theta}}(t) \right\|_2 \leq \frac{\|\mathbf{g}\|_2}{\lambda} e^{-t}. \tag{9}$$

By the fundamental theorem of calculus, we can express $\boldsymbol{\theta}(t)$ by integrating its gradient:

$$\begin{aligned} \boldsymbol{\theta}(t) &= \hat{\boldsymbol{\theta}} - \int_0^t e^{-\tau} \mathbf{H}_{\boldsymbol{\theta}(\tau)}^{-1} \mathbf{g} \mathrm{d}\tau \\ &= e^{-t} \hat{\boldsymbol{\theta}} + \int_0^t e^{-\tau} \big(\hat{\boldsymbol{\theta}} - \mathbf{H}_{\boldsymbol{\theta}(\tau)}^{-1} \mathbf{g} \big) \mathrm{d}\tau. \end{aligned} \tag{10}$$

For $0 < t < T$, the curve

$$\tau \mapsto \hat{\boldsymbol{\theta}} - \mathbf{H}_{\boldsymbol{\theta}(\tau)}^{-1} \mathbf{g}, \qquad 0 \leq \tau \leq t,$$

is continuous and takes values in $\mathcal{A}$ by assumption. Its image is compact, and the convex hull of this image is compact in $\mathbb{R}^d$ and contained in $\mathcal{A}$ by convexity of $\mathcal{A}$. Thus, the above integral is a nonnegative combination of points in $\mathcal{A}$, with weights given by $e^{-\tau}$.

Normalizing by the total weight to obtain a convex combination of points in $\mathcal{A}$, we get

$$\frac{1}{1 - e^{-t}} \int_0^t e^{-\tau} \big(\hat{\boldsymbol{\theta}} - \mathbf{H}_{\boldsymbol{\theta}(\tau)}^{-1} \mathbf{g} \big) \mathrm{d}\tau \in \mathcal{A} \,.$$

So, equation (10) expresses $\boldsymbol{\theta}(t)$ as a convex combination with weight e^{-t} on $\hat{\boldsymbol{\theta}}$. Because $\mathcal{A}$ is a neighborhood of $\hat{\boldsymbol{\theta}}$ and hence $\boldsymbol{\theta} \in \operatorname{int}(\mathcal{A})$, there exists $r > 0$ such that for all $t < T$, $\mathcal{A}$ contains a ball around $\boldsymbol{\theta}(t)$ of radius re^{-t}. Hence, for all $t < T$, $\boldsymbol{\theta}(t)$ is distance at least $re^{-T} > 0$ from the boundary of $\mathcal{A}$. But then we can apply Picard-Lindelöf at $\boldsymbol{\theta}(T)$ and extend the solution beyond T while staying in $\operatorname{int}(\mathcal{A})$, contradicting maximality of T. Hence $T = \infty$. □

Lemma D.3 (Convergence of $\boldsymbol{\theta}(t)$)**.** *The trajectory $\boldsymbol{\theta}(t)$ converges as $t \to \infty$ to a finite point $\boldsymbol{\theta}_\infty \in \mathbb{R}^d$.*

Proof. Lemma D.2 shows that the solution exists for all $t \geq 0$. For $0 \leq s < t$, equation (9) gives

$$\begin{aligned}\|\boldsymbol{\theta}(t) - \boldsymbol{\theta}(s)\|_2 &\leq \int_s^t \left\|\dot{\boldsymbol{\theta}}(\tau)\right\|_2 \mathrm{d}\tau \\ &\leq \frac{\|\mathbf{g}\|_2}{\lambda} \int_s^t e^{-\tau} \mathrm{d}\tau \\ &= \frac{\|\mathbf{g}\|_2}{\lambda}\big(e^{-s} - e^{-t}\big).\end{aligned}$$

Consequently, $\boldsymbol{\theta}(t)$ is Cauchy as $t \to \infty$ and converges to a finite point $\boldsymbol{\theta}_\infty \in \mathbb{R}^d$. □

Lemma D.4 (Convergence within $\mathcal{A}$)**.** *The limit $\boldsymbol{\theta}_\infty$ belongs to $\mathcal{A}$.*

Proof. Taking $t \to \infty$ in equation (10) gives

$$\boldsymbol{\theta}_\infty = \int_0^\infty e^{-\tau}\big(\hat{\boldsymbol{\theta}} - \mathbf{H}_{\boldsymbol{\theta}(\tau)}^{-1}\mathbf{g}\big)\mathrm{d}\tau.$$

The integral converges absolutely by the uniform bound $\left\|\mathbf{H}_{\boldsymbol{\theta}(\tau)}^{-1}\right\|_{\mathrm{op}} \leq 1/\lambda$, and its weights satisfy $\int_0^\infty e^{-\tau}\mathrm{d}\tau = 1$.

To show membership in $\mathcal{A}$, consider

$$C \coloneqq \mathrm{conv}\big\{\hat{\boldsymbol{\theta}} - \mathbf{H}_{\boldsymbol{\theta}(\tau)}^{-1}\mathbf{g} : \tau \geq 0\big\}.$$

The theorem's assumption and convexity imply $C \subseteq \mathcal{A}$. We clearly have $\boldsymbol{\theta}_\infty \in \overline{C}$. But suppose towards a contradiction that $\boldsymbol{\theta}_\infty \notin C$. The separating-hyperplane theorem, applied within the affine hull of C, gives an affine functional ℓ, nonconstant on that affine hull, such that

$$\ell(\boldsymbol{\theta}_\infty) = 0, \qquad \ell(x) \geq 0 \quad \text{for every } x \in C.$$

Since ℓ is affine and the weights integrate to one,

$$0 = \ell(\boldsymbol{\theta}_\infty) = \int_0^\infty e^{-\tau}\ell\big(\hat{\boldsymbol{\theta}} - \mathbf{H}_{\boldsymbol{\theta}(\tau)}^{-1}\mathbf{g}\big)\mathrm{d}\tau.$$

The integrand is continuous and nonnegative, and $e^{-\tau} > 0$. The vanishing integral therefore implies

$$\ell\big(\hat{\boldsymbol{\theta}} - \mathbf{H}_{\boldsymbol{\theta}(\tau)}^{-1}\mathbf{g}\big) = 0 \qquad \text{for every } \tau \geq 0.$$

Thus ℓ vanishes on C and hence on its affine hull, contradicting its nonconstancy there. We conclude that $\boldsymbol{\theta}_\infty \in C \subseteq \mathcal{A}$. □

Proof of Theorem D.1. By Lemmas D.3 and D.4, the Newton flow converges to a point $\boldsymbol{\theta}_\infty \in \mathcal{A}$. Continuity of the gradient and equation (8) give

$$\nabla L|_{\boldsymbol{\theta}_\infty} = \lim_{t\to\infty} \nabla L|_{\boldsymbol{\theta}(t)} = \lim_{t\to\infty} e^{-t}\mathbf{g} = \mathbf{0}.$$

Since $\boldsymbol{\theta}_\infty \in \mathcal{A}$, the Hessian satisfies

$$\nabla^2 L|_{\boldsymbol{\theta}_\infty} = \mathbf{H}_{\boldsymbol{\theta}_\infty} \succ \lambda\mathbf{I}.$$

The second-order sufficient condition therefore shows that $\boldsymbol{\theta}_\infty$ is a strict local minimum of L.

If L is also strongly convex on all of $\mathbb{R}^d$, this stationary point is its unique global minimizer. Consequently,

$$\boldsymbol{\theta}^\star \coloneqq \operatorname*{argmin}_{\boldsymbol{\theta}\in\mathbb{R}^d} L(\boldsymbol{\theta}) = \boldsymbol{\theta}_\infty \in \mathcal{A}\,.$$ □

E Third Order Tensors

A key difficulty in using approximation-error bounds such as those of (Sekhari et al., 2021; Rubinstein & Hopkins, 2026) for certified unlearning is their dependence on how quickly the Hessian of the loss can change. A Newton step uses the curvature at one model to approximate the loss nearby, and changes in this curvature control the resulting approximation error. To turn this into an unlearning guarantee, the masking step needs a concrete numerical bound on that error, and a loose bound on how quickly the Hessian changes can translate into unnecessarily large amounts of masking noise.

From Hessian Lipschitzness to third derivatives. The relevant measure of change is the Lipschitz constant of the Hessian with respect to the *operator norm*. For a three-times continuously differentiable loss L, this constant is

$$C_{\mathrm{Lip}} \coloneqq \sup_{\substack{\boldsymbol{\theta}\in\mathbb{R}^d \\ \mathbf{h}\in\mathbb{R}^d\setminus\{\mathbf{0}\}}} \frac{\left\|\nabla^2 L(\boldsymbol{\theta}+\mathbf{h}) - \nabla^2 L(\boldsymbol{\theta})\right\|_{\mathrm{op}}}{\|\mathbf{h}\|_2} = \sup_{\boldsymbol{\theta}\in\mathbb{R}^d} \left\|\nabla^3 L(\boldsymbol{\theta})\right\|_{\mathrm{op}}. \tag{11}$$

The equality is the higher-dimensional analogue of bounding the Lipschitz constant of a scalar function by the magnitude of its derivative.

Third derivatives for logistic regressions. For a logistic regression with samples $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$, we may write the loss as[6]

$$L(\boldsymbol{\theta}) = \sum_{i=1}^{n} \ell(\langle \boldsymbol{\theta}, \mathbf{x}_i\rangle; y_i).$$

Differentiating the loss three times gives

$$\mathbf{T}(\boldsymbol{\theta}) \coloneqq \nabla^3 L(\boldsymbol{\theta}) = \sum_{i=1}^{n} \ell'''(\langle \boldsymbol{\theta}, \mathbf{x}_i\rangle; y_i)\, \mathbf{x}_i \otimes \mathbf{x}_i \otimes \mathbf{x}_i,$$

where ℓ''' denotes differentiation with respect to the scalar prediction argument. Therefore, the third derivative separates into a scalar contribution from the loss and a geometric contribution from each covariate.

For binary logistic regression, using labels $y \in \{0, 1\}$, the loss is $\ell(z; y) = \log(1 + e^z) - yz$, and

$$\ell'''(z; y) = \sigma(z)(1 - \sigma(z))(1 - 2\sigma(z)) \in \left[\frac{-1}{6\sqrt{3}}, \frac{1}{6\sqrt{3}}\right],$$

where $\sigma(z) \coloneqq \frac{1}{1+e^{-z}}$ is the sigmoid function. Note that this third derivative is independent of the label.

Because $\mathbf{T}(\boldsymbol{\theta})$ is symmetric, its operator norm can be evaluated using the same unit direction in all three arguments (Banach's theorem; see e.g. Theorem 6.9 of (Hillar & Lim, 2013)):

$$\|\mathbf{T}(\boldsymbol{\theta})\|_{\mathrm{op}} = \sup_{\|\mathbf{h}\|_2=1} |\mathbf{T}(\boldsymbol{\theta})[\mathbf{h}, \mathbf{h}, \mathbf{h}]| = \sup_{\|\mathbf{h}\|_2=1} \left|\sum_{i=1}^{n} \ell'''(\langle \boldsymbol{\theta}, \mathbf{x}_i\rangle; y_i)\langle \mathbf{x}_i, \mathbf{h}\rangle^3\right|.$$

In other words, even at a fixed model $\boldsymbol{\theta}$, computing $\left\|\nabla^3 L|_{\boldsymbol{\theta}}\right\|_{\mathrm{op}}$ implicitly requires maximizing a degree-3 polynomial over the unit sphere. Unlike computing a matrix spectral norm, this problem is NP-hard for general symmetric tensors (Hillar & Lim, 2013), and leading approximation guarantees lose polynomial factors in the dimension (Bhattiprolu et al., 2017).

An efficiently computable upper bound. For certification, we do not need the exact tensor norm and a valid upper bound suffices, and we describe such a bound based on (Hopkins et al., 2015) here.

Suppose that the scalar loss admits a finite uniform bound

$$\kappa_3 \geq \sup_{z,y} |\ell'''(z; y)|.$$

For logistic regression, maximizing the expression above over $\sigma(z) \in (0, 1)$ gives $\kappa_3 = 1/(6\sqrt{3})$. Then, for any unit direction $\mathbf{h}$, Cauchy–Schwarz gives

$$|\mathbf{T}(\boldsymbol{\theta})[\mathbf{h}, \mathbf{h}, \mathbf{h}]| \leq \kappa_3 \sum_{i=1}^{n} |\langle \mathbf{x}_i, \mathbf{h}\rangle|^3 \leq \kappa_3 \left(\sum_{i=1}^{n} \langle \mathbf{x}_i, \mathbf{h}\rangle^2\right)^{1/2} \left(\sum_{i=1}^{n} \langle \mathbf{x}_i, \mathbf{h}\rangle^4\right)^{1/2}.$$

[6] In cases where a quadratic regularizer is added, that does not change C_{Lip}, since its Hessian is constant.

Algorithm 4: Gram-matrix bound on the Hessian Lipschitz constant

Input: Data matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ with rows $\mathbf{x}_i^\intercal$; derivative bound $\kappa_3 \geq \sup_{z,y} |\ell'''(z; y)|$.

Output: An upper bound $\widehat{C_{\text{Lip}}}$ on the Hessian Lipschitz constant of the summed loss.

$\mathbf{G} \leftarrow \mathbf{X}\mathbf{X}^\intercal$

$\mathbf{G}_2 \leftarrow \mathbf{G} \odot \mathbf{G}$ `// Entrywise square`

$s_1 \leftarrow \sigma_{\max}(\mathbf{G})$

$s_2 \leftarrow \sigma_{\max}(\mathbf{G}_2)$

return $\kappa_3 \sqrt{s_1 s_2}$

This separates the cubic expression into a quadratic term and a quartic term. The quadratic term is controlled directly by the data matrix. For the quartic term, we can obtain a matrix bound by treating each pair of feature coordinates as a new feature.

Concretely, let $\mathbf{X} \in \mathbb{R}^{n \times d}$ have rows $\mathbf{x}_i^\intercal$, and let $\mathbf{G} \coloneqq \mathbf{X}\mathbf{X}^\intercal$ be its Gram matrix, so that $\mathbf{G}_{ij} = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$. Also consider the lifted matrix $\mathbf{X}_2 \in \mathbb{R}^{n \times d^2}$ with rows $(\mathbf{x}_i \otimes \mathbf{x}_i)^\intercal$. Its Gram matrix is $\mathbf{X}_2\mathbf{X}_2^\intercal = \mathbf{G}^{\odot 2}$, where $\odot 2$ denotes entrywise squaring, since $\langle \mathbf{x}_i \otimes \mathbf{x}_i, \mathbf{x}_j \otimes \mathbf{x}_j \rangle = \langle \mathbf{x}_i, \mathbf{x}_j \rangle^2$. Consequently,

$$\sum_{i=1}^{n} \langle \mathbf{x}_i, \mathbf{h} \rangle^2 = \|\mathbf{X}\mathbf{h}\|_2^2 \leq \sigma_{\max}(\mathbf{G}),$$

$$\sum_{i=1}^{n} \langle \mathbf{x}_i, \mathbf{h} \rangle^4 = \|\mathbf{X}_2(\mathbf{h} \otimes \mathbf{h})\|_2^2 \leq \sigma_{\max}\left(\mathbf{G}^{\odot 2}\right),$$

where the second inequality uses $\|\mathbf{h} \otimes \mathbf{h}\|_2 = \|\mathbf{h}\|_2^2 = 1$. Taking the supremum over $\mathbf{h}$ and $\boldsymbol{\theta}$ therefore yields

$$C_{\text{Lip}} \leq \widehat{C_{\text{Lip}}} \coloneqq \kappa_3 \sqrt{\sigma_{\max}(\mathbf{G})\ \sigma_{\max}(\mathbf{G}^{\odot 2})}. \tag{12}$$

The bound reflects how strongly the samples can accumulate along a common direction. For example, for unit-norm covariates it equals κ_3 when the covariates are mutually orthogonal, but $n\kappa_3$ when they are all identical. However, it need not be tight: taking absolute values loses cancellation, and the lifted matrix bound optimizes over all unit vectors rather than only vectors of the form $\mathbf{h} \otimes \mathbf{h}$. Thus, obtaining a computable upper bound does not by itself resolve the difficulty of calibrating masking noise accurately. For numerical certification, the spectral-norm computations must also use verified upper bounds rather than estimates that could understate the largest eigenvalues.