When to Forget? Complexity Trade-offs in Machine Unlearning

Martin Van Waerebeke [1] **Marco Lorenzi** [2] **Giovanni Neglia** [2] **Kevin Scaman** [1]

Abstract

Machine Unlearning (MU) aims at removing the influence of specific data points from a trained model at a fraction of the cost of full model retraining. In this paper, we analyze the efficiency of unlearning methods and establish the first upper and lower bounds on minimax computation times for this problem, characterizing the performance of the most efficient algorithm against the most difficult objective function. Specifically, for strongly convex objective functions and under the assumption that the forget data is inaccessible to the unlearning method, we provide a phase diagram for the unlearning complexity ratio—a novel metric that compares the computational cost of the best unlearning method to full model retraining. The phase diagram shows three regimes: one where unlearning is too costly, one where it's trivial, and one where it outperforms retraining. These findings highlight the critical role of factors such as data dimensionality, the number of samples to forget, and privacy constraints in determining the practical feasibility of unlearning.

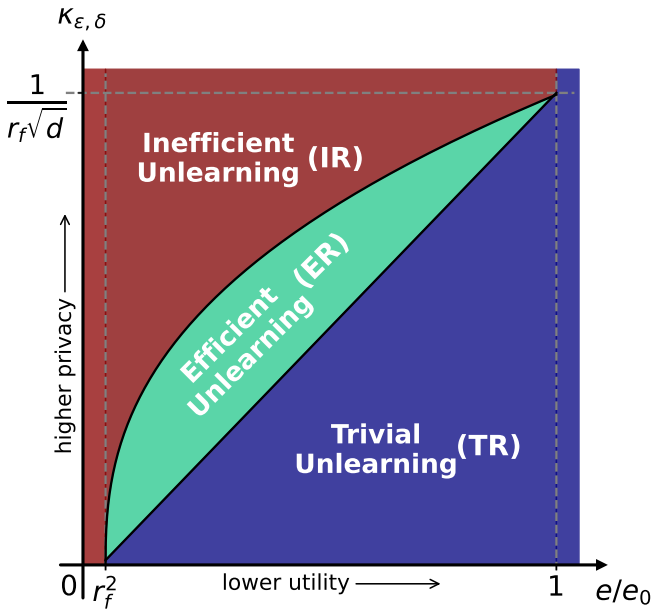

Figure 1. Schematic representation of the phase diagram for unlearning, where e (resp. e_0) is the target (resp. initial) excess risk, and $\kappa_{\epsilon,\delta}$ the strength of the privacy constraint (see Section 4). We describe the existence of three regimes of unlearning (IR, ER, TR).

1. Introduction

The growing use of personal data in machine learning raises privacy concerns under regulations such as GDPR and CCPA (Mantelero, 2013; Goldman, 2020). The obligation to allow for individual data erasure may require retraining models from scratch, which is costly (Cottier et al., 2024).

To mitigate this cost, Machine Unlearning (MU) aims to remove the influence of specific training data—the "forget set"— at a fraction of the cost of full retraining (Kong and Chaudhuri, 2023). However, this "fraction" is not quantified in the general literature. We thus propose to study the *unlearning complexity ratio*: the ratio between the number of steps needed for unlearning and retraining, denoted T_e^U/T_e^S for a target excess risk e. This key measure of unlearning efficiency evaluates the ability for unlearning algorithms to reduce computational cost as compared to retraining. Establishing general bounds on this ratio is crucial for ensuring the efficiency of MU, and is the subject of this paper.

Contributions. In this work, we :

- Introduce the *unlearning complexity ratio* (unlearning time over retraining time), leveraging minimax optimization complexity in MU for the first time.
- Provide the first lower bound for unlearning complexityeu, answering an open problem in the literature (Allouah et al., 2024) and identifying a regime of inefficiency for unlearning (IR).
- Derive the first upper bound for this ratio, exhibiting a regime in which unlearning is provably faster than retraining (ER).
- Identify a last regime where unlearning is trivial (TR).

[1]INRIA Paris [2]INRIA Sophia Antipolis. Correspondence to: Martin Van Waerebeke <martin.van-waerebeke@inria.fr>.

Proceedings of the 42nd International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

2. Related Work

Machine Unlearning (MU) is an emerging research field focused on removing the influence of specific data from trained models. A key distinction exists between empirical methods, which aim for efficiency but offer no guarantees (Kurmanji et al., 2024), and certified methods, which provide formal assurances of successful unlearning.

Certified unlearning methods fall into two main categories: exact and approximate. Exact approaches, while offering stronger guarantees, often require changes to the training process, such as sharding (Bourtoule et al., 2021) or tree-based structures (Ullah and Arora, 2023). In contrast, approximate methods balance utility and data removal. Many of these approaches (Neel et al., 2021; Fraboni et al., 2024) are built on the framework of Differential Privacy (DP) (Dwork and Roth, 2014), which ensures robustness against privacy attacks such as membership inference.

Comparisons with retraining from scratch are common in the literature, as unlearning should be faster. However, such comparisons are often empirical and lack theoretical support (Hayes et al., 2024). Some recent studies have started addressing this gap by analyzing computational costs and utility trade-offs of different unlearning strategies (Izzo et al., 2021; Chourasia and Shah, 2023; Allouah et al., 2024), with differences in assumptions and goals.

3. Problem Setup

3.1. Learning and Unlearning Setups

Consider a supervised learning setting on a dataset $\mathcal{D}$ in which our goal is to minimize the objective function

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d} \mathcal{L}(\boldsymbol{\theta}) \coloneqq \mathbb{E}\left[\ell(\boldsymbol{\theta},\xi)\right] , \tag{1}$$

where $\ell : \mathbb{R}^d \times \mathbb{R}^s \to \mathbb{R}$ is a loss function, $\boldsymbol{\theta} \in \mathbb{R}^d$ the vector of model parameters, and $\xi \sim \mathcal{D}$ the data. For a fraction $r_f \in [0,1]$, we decompose our distribution as follows,

$$\mathcal{D} \coloneqq r_f\, \mathcal{D}_f \;+\; \left(1 - r_f\right) \mathcal{D}_r \,, \tag{2}$$

where $\mathcal{D}_f$ is the distribution one wishes to remove (*i.e.,* , the *forget* distribution), while $\mathcal{D}_r$ is the distribution that is unaffected by the removal (*i.e.,* , the *retain* distribution).

Assumption 1 (loss regularity)**.** *Let $\mu, L > 0$, and $R = L/2\mu$. For any $\xi \in \mathbb{R}^s$, the loss function $\ell(\cdot,\xi)$ of Eq. (1) is L-Lipschitz and μ-strongly convex on $\mathbb{B}(0,R)$.*

Although not directly applicable to most neural networks, these assumptions or even stronger counterparts have been used in the literature (Huang and Canonne, 2023; Allouah et al., 2024). We denote as $\mathcal{F}_{sc}(\mu, L)$ the class of such loss functions, abbreviated to $\mathcal{F}_{sc}$ when there is no ambiguity.

3.2. Iterative First-Order Algorithms

In this, section, we provide intuitive definitions for learning and unlearning algorithms. More rigorous definitions are available in Appendix E. We will consider that both types of algorithms are *non-deterministic*, *iterative* and *first-order*. The set of learning (resp. (ϵ,δ)-unlearning, see Def 1) algorithms are referred to as $\mathbb{A}$ (resp. $\mathbb{U}_{\epsilon,\delta}$).

For a learning (resp. unlearning) algorithm $\mathcal{A} \in \mathbb{A}$ (resp. $\mathcal{U} \in \mathbb{U}_{\epsilon,\delta}$), we denote as $\mathcal{A}(T,\ell,\mathcal{D}_r)$ (resp. $\mathcal{U}(T,\ell,\mathcal{D}_r,\mathcal{D}_f)$) the output of the learning (resp. unlearning) algorithm iterated T times on the loss ℓ with dataset $\mathcal{D}_r$. In the case of the learning algorithm, the initialization is taken at random, whereas for the unlearning algorithm it is taken as the unique minimizer of the loss ℓ on $\mathcal{D}_r \cup \mathcal{D}_f$.

3.3. Unlearning Guarantees

Unlearning aims at removing the impact of the forget set on the trained model. We use a slightly modified version of the DP-based definition introduced in Ginart et al. (2019).

Definition 1 ((ϵ,δ)-Unlearning)**.** *An unlearning algorithm $\mathcal{U} \in \mathbb{U}_{\epsilon,\delta}$ satisfies (ϵ,δ)-Unlearning, if, for any triplet of distributions $(\mathcal{D}_r, \mathcal{D}_f, \mathcal{D}'_f)$, loss function ℓ, and for any subset of outputs $S \subset \mathbb{R}^d$, the following holds,*

$$\mathbb{P}[\mathcal{U}(\mathcal{D}_r,\mathcal{D}_f) \in S] \le e^{\epsilon} \cdot \mathbb{P}[\mathcal{U}(\mathcal{D}_r,\mathcal{D}'_f) \in S] + \delta.$$

The values (ϵ,δ) are called the unlearning budget, where a low budget means harder unlearning. We define $\kappa_{\epsilon,\delta} \coloneqq \epsilon^{-1}\sqrt{2\,\ln(1.25/\delta)}$ as the strength of the privacy constraint.

3.4. Minimax Computation Times

Let us start by introducing some key elements. First, we define the time required to re-learn from scratch and to unlearn. For a given excess risk threshold e, loss ℓ and algorithms $\mathcal{A} \in \mathbb{A}$ and $\mathcal{U} \in \mathbb{U}_{\epsilon,\delta}$, one can define the time needed to get an excess risk smaller than e as

$$T_e^S(\ell,\mathcal{A}) \coloneqq \min_{T\in\mathbb{N}}\{T;\ \mathbb{E}[\mathcal{L}_r(\mathcal{A}(T,\ell,\mathcal{D}_r)) - \mathcal{L}_r^*] \le e\} \,,$$

$$T_e^U(\ell,\mathcal{U}) \coloneqq \min_{T\in\mathbb{N}}\{T;\ \mathbb{E}[\mathcal{L}_r(\mathcal{U}(T,\ell,\mathcal{D}_r,\mathcal{D}_f)) - \mathcal{L}_r^*] \le e\} \,.$$

where $\mathcal{L}_r(\boldsymbol{\theta}) = \mathbb{E}_{\xi\sim\mathcal{D}_r}\left[\ell(\boldsymbol{\theta},\xi)\right]$.

When studying the performance of an algorithm over a function class, one wants to study the worst-case performance of any algorithm $\mathcal{A}$ and to find the algorithm minimizing this worst case. Therefore, one can define the minimax retraining time of algorithms in $\mathbb{A}$ over the class $\mathcal{F}_{sc}$ as

$$T_e^S \coloneqq \inf_{\mathcal{A}\in\mathbb{A}} \sup_{\ell\in\mathcal{F}_{sc}} T_e^S(\ell,\mathcal{A}). \tag{3}$$

In the same way, we define the minimax forget time of unlearning algorithms in $\mathbb{U}_{\epsilon,\delta}$ over the class $\mathcal{F}_{sc}$ as

$$T_e^U \coloneqq \inf_{\mathcal{U}\in\mathbb{U}_{\epsilon,\delta}} \sup_{\ell\in\mathcal{F}_{sc}} T_e^U(\ell,\mathcal{U}). \tag{4}$$

4. Regimes of Unlearning Complexity

In this section, we provide lower and upper bounds for the *unlearning complexity ratio* T_e^U/T_e^S. By doing so, we identify regimes in which first-order unlearning methods can-or cannot-be significantly faster than retraining.

4.1. Speed of Retraining from Scratch

We start with some preliminary results. In order for our lower bounds to hold, we need a technical assumption, which usually holds for continuous distributions, as long as $\text{supp}(\mathcal{D}_r)$ and $\text{supp}(\mathcal{D}_f)$ do not cover the whole space $\mathbb{R}^s$.

Assumption 2. *(Flexible distributions) For any $p \in [0,1]$, $\exists A \subset \mathbb{R}^s$ s.t. $\mathbb{P}(\xi_r \in A) = p$, where $\xi_r \sim \mathcal{D}_r$. Moreover, there exists a distribution $\mathcal{D}'_f$ such that supp$(\mathcal{D}_r)$, supp$(\mathcal{D}_f)$ and supp$(\mathcal{D}'_f)$ are two-by-two disjoint.*

Learning is trivial (*i.e.*, $T_e^S = 0$) if $e \geq e_0 := \frac{L^2}{8\mu}$, as $\boldsymbol{\theta}_0 = 0$ already satisfies the target excess risk. Additionally, when it is not, learning speed is well-known: under Assumption 2, and if $e < e_0$, we have $T_e^S = \Theta\left(\frac{e_0}{e}\right)$. To claim that an unlearning method is efficient will thus require it to have complexity under $O(e_0/e)$. We will show that such unlearning algorithms do exist in Section 4.4.

4.2. Trivial Unlearning Regime

We now start with the simplest case: for a high target excess risk e and low privacy constraint $\kappa_{\epsilon,\delta}$, simply adding Gaussian noise to the parameters of the model is sufficient.

Theorem 1 (Trivial regime)**.** *If the target excess risk verifies $e \in \left[\frac{r_f}{1-r_f}\left(\frac{r_f}{1-r_f} + \sqrt{d}\kappa_{\epsilon,\delta}\right)e_0, e_0\right)$, then*

$$\frac{T_e^U}{T_e^S} = 0\,. \tag{5}$$

This first regime corresponds to the blue area in Figure 1, where unlearning can be performed with 0 gradient access.

4.3. Impossible Unlearning Regime

Conversely, we now show the existence of a regime in which unlearning cannot asymptotically outperform retraining.

Theorem 2 (Impossible regime)**.** *Let $\delta \in [10^{-8}, \epsilon]$. Under Assumption 2, there exists a universal constant $c > 0$ such that, if $e < \min\left\{1,\, c\left(\frac{r_f}{1-r_f}\right)^2\left(1+\kappa_{\epsilon,\delta}^2\right)\right\}e_0$, then*

$$\frac{T_e^U}{T_e^S} = \Omega(1)\,. \tag{6}$$

Theorem 2 provides a regime in which first-order unlearning methods cannot asymptotically outperform retraining. This regime is delimited by a curve of type $\kappa_{\epsilon,\delta} \geq \alpha\sqrt{e}$, with α a constant, explaining our choice of representation in Fig. 1. For low forget ratios $r_f \ll 1$, the unlearning complexity ratio is lower bounded by a constant when e is below a quantity proportional to $r_f^2(1+\kappa_{\epsilon,\delta}^2)e_0$. In this regime, removing even minimal parts of a dataset requires a non-negligible retraining time. This may be an issue when numerous small removals must be made to a model, as each of these removals will incur a cost proportional to that of its full retraining. Finally, when the target excess risk e tends to 0, a direct corollary of Theorem 2 is that unlearning cannot asymptotically outperform retraining, regardless of the strength of the privacy constraint $\kappa_{\epsilon,\delta}$.

Corollary 1. *Within the hypothesis of Theorem 2, and for $r_f \in (0,1)$ and $\kappa_{\epsilon,\delta} \geq 0$ fixed, we have*

$$\liminf_{e\to 0} \frac{T_e^U}{T_e^S} > 0\,. \tag{7}$$

In other words, the advantage of starting from $\boldsymbol{\theta}^*$ rather than a random model reduces as e decreases, and until unlearning cannot asymptotically outperform retraining .

4.4. Efficient Unlearning Regime

We have now identified that unlearning is trivial on one end of the spectrum, while it is inefficient on the other. We now characterize what happens between those two extremes by showing that a simple unlearning mechanism achieves a good unlearning complexity ratio in this intermediate regime. To do so, we derive an upper bound on unlearning time using the unlearning algorithm "noise and fine-tune" (see Algorithm 2), which is an adapted version of Neel et al. (2021)'s perturbed gradient descent. Using known learning convergence speeds, such a bound implies an upper bound on the unlearning complexity ratio.

Theorem 3 (Efficient regime)**.** *For any $e < e_0$, we have*

$$\frac{T_e^U}{T_e^S} = \mathcal{O}\left(\left(\frac{r_f}{1-r_f}\right)^2\left(1+d\kappa_{\epsilon,\delta}^2\right)\frac{e_0}{e}\right)\,. \tag{8}$$

Theorem 3 showcases the possibility of efficient unlearning, with an unlearning complexity ratio proportional to r_f^2. For low forget ratios $r_f \ll 1$, the "noise and fine-tune" method outperforms retraining (*i.e.*, $T_e^U < T_e^S$) when the target excess risk is above a quantity proportional to $r_f^2\left(1+d\kappa_{\epsilon,\delta}^2\right)e_0$, and we recover, up to a constant and for a fixed dimension d, the regime in which unlearning becomes possible in Theorem 2 (see Section 4.3). The combination of both Theorem 2 and Theorem 3 thus shows that $r_f^2\left(1+d\kappa_{\epsilon,\delta}^2\right)e_0$ acts as a threshold for the target excess risk before which efficient unlearning is impossible, and above which unlearning becomes efficient (and even trivial beyond $r_f(r_f+\sqrt{d}\kappa_{\epsilon,\delta})e_0$).

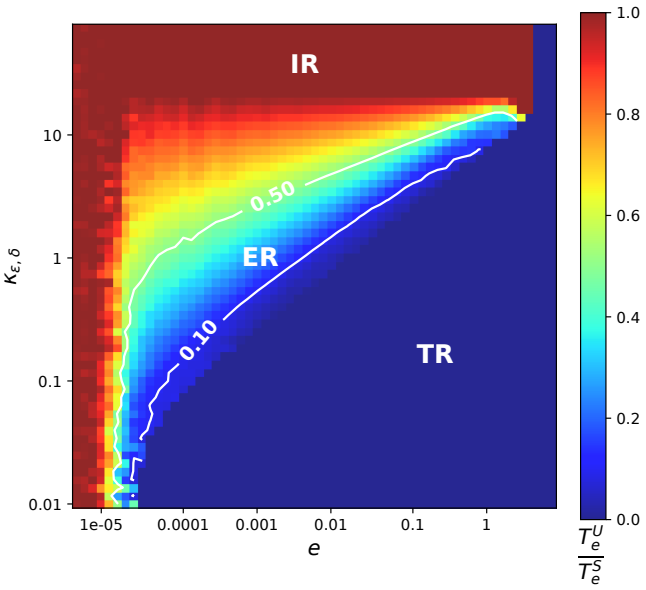

Figure 2. Experimental phase diagram of the unlearning complexity ratio. We give estimates for T_e^U and T_e^S using the "noise and fine-tune" (Algorithm 2) and SGD algorithms, respectively. We display the value of their ratio as a function of $\kappa_{\epsilon,\delta}$ and e in log-log scale. We notice the three regimes described in our theoretical analysis: impossible (IR), efficient (ER), and trivial (TR).

4.5. Discussion

Overall, our analysis shows that there are three main regimes—Trivial, Impossible, and Efficient—that describe how unlearning time compares to retraining time, based on the target excess risk e, the strength of the privacy constraint $\kappa_{\epsilon,\delta}$ and the forget ratio r_f. Figure 1 illustrates these regimes and their boundaries.

Since we rely on noising the model parameters to ensure unlearning, our bound scales with $\sqrt{d}$, as is common in differentially-private optimization (Bassily et al., 2014). While natural, this dependence is not matched by our lower bound in Theorem 2. We leave the exploration of this discrepancy to future work.

5. Experiments

We experimentally investigate the landscape of the unlearning complexity ratio as a function of e and $\kappa_{\epsilon,\delta}$.

5.1. Experimental Setting

In order to give an estimate of the unlearning complexity ratio, we need to choose specific algorithms to represent the learning algorithm class $\mathbb{A}$ as well as the unlearning algorithm class $\mathbb{U}_{\epsilon,\delta}$. For the learning algorithm, we choose the canonical stochastic gradient descent, as defined in (Garrigos and Gower, 2023). For the unlearning algorithm, we choose the "noise and fine-tune" algorithm (see Alg. 2)-the one used to dervie the bound in Theorem 3.

We compare the learning and unlearning algorithms on the Digit dataset, a subset of Alpaydin and Alimoglu (1996), across various values of e and $\kappa_{\epsilon,\delta}$. Using logistic regression with cross-entropy loss and $L2$ regularization, we train until the error threshold e is met, starting from random initialization (learning) or a noised version of the optimum (unlearning). We report the ratio of unlearning to retraining time as a proxy for unlearning complexity.

5.2. Experimental Results

Figure 2 illustrates the empirical unlearning complexity ratio. It displays the three regimes described in our theoretical analysis. As expected, a high privacy budget coupled with a permissive e allow for an immediate convergence of unlearning. Additionally, taking e too small inevitably prevents unlearning from outperforming retraining, regardless of $\kappa_{\epsilon,\delta}$.

6. Conclusion

In this paper, we study the efficiency of machine unlearning through the lens of a novel metric —the *unlearning complexity ratio*— which compares the worst-case convergence speeds of the best unlearning and retraining algorithms. Our analysis reveals three regimes. In one (TR), we show that unlearning can be done "for free" (Theorem 1). In another (IR), described by our lower bound on the unlearning complexity ratio (Theorem 2), unlearning cannot asymptotically beat retraining through gradient-based methods. In the last regime (ER), our upper bound on the unlearning complexity ratio shows that unlearning is possible at a small fraction of the cost of retraining, a cost that scales with the square of the fraction of forgotten samples (Theorem 3).

Empirical validation confirms these insights, showing the utility of analysing unlearning through the minimax complexity framework. Beyond unveiling fundamental limits and opportunities, our results address an open question on whether unlearning can outperform retraining—and under what circumstances. We introduce the first bounds on the unlearning complexity ratio, as well as the first lower bound on unlearning time.

We hope that the framework and findings presented here will stimulate further studies on machine unlearning, and allow further analysis of a wider class of algorithms, objective functions, and data distributions. Specifically, lower-bounding the unlearning complexity ratio for methods beyond the first order, relying on the forget set, or not verifying Assumption 2 remains an open challenge.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning and Unlearning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

REFERENCES

Allouah, Y., Kazdan, J., Guerraoui, R., and Koyejo, S. (2024). The utility and complexity of in-and out-of-distribution machine unlearning. *arXiv preprint arXiv:2412.09119*.

Alpaydin, E. and Alimoglu, F. (1996). Pen-Based Recognition of Handwritten Digits. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5MG6K.

Bassily, R., Smith, A., and Thakurta, A. (2014). Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th annual symposium on foundations of computer science*, pages 464–473. IEEE.

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

Bubeck, S. et al. (2015). Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357.

Chourasia, R. and Shah, N. (2023). Forget unlearning: Towards true data-deletion in machine learning. In *International Conference on Machine Learning*, pages 6028–6073. PMLR.

Cottier, B., Rahman, R., Fattorini, L., Maslej, N., and Owen, D. (2024). The rising costs of training frontier ai models. *arXiv preprint arXiv:2405.21015*.

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

Fraboni, Y., Van Waerebeke, M., Scaman, K., Vidal, R., Kameni, L., and Lorenzi, M. (2024). Sifu: Sequential informed federated unlearning for efficient and provable client unlearning in federated optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 3457–3465. PMLR.

Garrigos, G. and Gower, R. M. (2023). Handbook of convergence theorems for (stochastic) gradient methods. *arXiv preprint arXiv:2301.11235*.

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

Goldman, E. (2020). An introduction to the california consumer privacy act (ccpa). *Santa Clara Univ. Legal Studies Research Paper*.

Hayes, J., Shumailov, I., Triantafillou, E., Khalifa, A., and Papernot, N. (2024). Inexact unlearning needs more careful evaluations to avoid a false sense of privacy. *arXiv preprint arXiv:2403.01218*.

Huang, Y. and Canonne, C. L. (2023). Tight bounds for machine unlearning via differential privacy. *arXiv preprint arXiv:2309.00886*.

Izzo, Z., Anne Smart, M., Chaudhuri, K., and Zou, J. (2021). Approximate data deletion from machine learning models. In Banerjee, A. and Fukumizu, K., editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 2008–2016. PMLR.

Kong, Z. and Chaudhuri, K. (2023). Data redaction from pre-trained gans. In *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pages 638–677. IEEE.

Kurmanji, M., Triantafillou, P., Hayes, J., and Triantafillou, E. (2024). Towards unbounded machine unlearning. *Advances in neural information processing systems*, 36.

Mantelero, A. (2013). The eu proposal for a general data protection regulation and the roots of the 'right to be forgotten'. *Computer Law & Security Review*, 29(3):229–235.

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

Ullah, E. and Arora, R. (2023). From adaptive query release to machine unlearning. In *International Conference on Machine Learning*, pages 34642–34667. PMLR.

Villani, C. et al. (2009). *Optimal transport: old and new*, volume 338. Springer.

A. Upper bounds

Proof of Theorem 1. Let $e \geq \frac{r_f}{1-r_f}\left(\frac{r_f}{1-r_f} + \sqrt{d}\kappa_{\epsilon,\delta}\right) e_0$.

Let the unlearning algorithm consist in simply adding Gaussian noise to the previous optimum $\boldsymbol{\theta}^*$. By application of the Gaussian mechanism (Dwork and Roth, 2014), adding i.i.d. Gaussian noise with standard deviation $\kappa_{\epsilon,\delta} \|\boldsymbol{\theta}_r^* - \boldsymbol{\theta}^*\|$ ensure (ϵ, δ)-Unlearning of the forget set. Using the bound from Lemma C.1, we sample:

$$g \sim \mathcal{N}(0, (\kappa_{\epsilon,\delta} \frac{r_f}{1-r_f} \frac{L}{\mu})^2 I_d), \tag{9}$$

where I_d is the identity matrix in $\mathbb{R}^d$.

Let $\hat{\boldsymbol{\theta}} := \boldsymbol{\theta}^* + g$. We can then bound the expected loss of $\hat{\boldsymbol{\theta}}$:

$$\begin{aligned}
\mathbb{E}\left[\mathcal{L}_r(\hat{\boldsymbol{\theta}}) - \mathcal{L}_r^*\right] &\leq \mathbb{E}\left[\mathcal{L}_r(\hat{\boldsymbol{\theta}}) - \mathcal{L}_r(\boldsymbol{\theta}^*)\right] + \mathcal{L}_r(\boldsymbol{\theta}^*) - \mathcal{L}_r^* && (10)\\
&\leq \sqrt{d}\kappa_{\epsilon,\delta}\left(\frac{r_f}{1-r_f}\right)^2 \frac{L^2}{\mu} + \frac{r_f}{1-r_f}\frac{L^2}{\mu} && (11)\\
&\leq r_f\left(r_f + \sqrt{d}\kappa_{\epsilon,\delta}\right) e_0 && (12)\\
&\leq e\,. && (13)
\end{aligned}$$

□

Proof of Theorem 3. According to Lemma C.1, we have

$$\|\boldsymbol{\theta}^* - \boldsymbol{\theta}_r^*\| \leq \frac{r_f}{1-r_f}\frac{L}{\mu} =: R_1\,. \tag{14}$$

To perform the unlearning, we use the noise + fine-tune method as introduced in Algorithm 2. For the noising part, the standard deviation of the noise that needs to be added to ensure (ϵ, δ)-Unlearning of $\mathcal{D}_f$ is $\kappa_{\epsilon,\delta} \|\boldsymbol{\theta}^* - \boldsymbol{\theta}_r^*\|$. Thus, we set $\sigma = \kappa_{\epsilon,\delta} R_1$ and define $\widetilde{\boldsymbol{\theta}} := \boldsymbol{\theta}^* + g$, where $g \sim \mathcal{N}(0, \sigma^2)$.

Now, one can notice that

$$\mathbb{E}\left[\left\|\widetilde{\boldsymbol{\theta}} - \boldsymbol{\theta}_r^*\right\|^2\right] = \mathbb{E}\left[\|g\|^2\right] + \|\boldsymbol{\theta}^* - \boldsymbol{\theta}_r^*\|^2 \leq (1 + d\kappa_{\epsilon,\delta}^2) R_1^2\,. \tag{15}$$

While we do not know the exact distance $\left\|\widetilde{\boldsymbol{\theta}} - \boldsymbol{\theta}_r^*\right\|$ our SGD will need to cover, we have its expectation. Thus, we set the learning rate γ to be optimal for a the expectation of the distance, *i.e.,* : $\gamma = \sqrt{\frac{(1+d\kappa_{\epsilon,\delta}^2)R_1^2}{T^U L^2}}$.

Let $\mathcal{A}_\gamma$ be the SSD algorithm with learning rate γ, as defined in Section 3 of Garrigos and Gower (2023). Using Theorem 9.7 from Garrigos and Gower (2023), we get

$$\begin{aligned}
\mathbb{E}\left[\mathcal{L}_r(\mathcal{A}_\gamma(\widetilde{\boldsymbol{\theta}}, \mathcal{D}_r, T)) - \mathcal{L}_r^*\right] &\leq \mathbb{E}\left[\frac{\left\|\widetilde{\boldsymbol{\theta}} - \boldsymbol{\theta}_r^*\right\|^2}{2\gamma T} + \frac{\gamma L^2}{2}\right] && (16)\\
&\leq \frac{(1 + d\kappa_{\epsilon,\delta}^2)R_1^2}{2\gamma T} + \frac{\gamma L^2}{2} && (17)\\
&\leq \frac{LR_1}{\sqrt{T}}\sqrt{1 + d\kappa_{\epsilon,\delta}^2}\,. && (18)
\end{aligned}$$

For a given excess risk threshold e, the unlearning time can then be upper-bounded as

$$T_e^U \leq \frac{L^2 R_1^2}{e^2}(1 + d\kappa_{\epsilon,\delta}^2) = \left(\frac{r_f}{1 - r_f}\right)^2 (1 + d\kappa_{\epsilon,\delta}^2) \left(\frac{e_0}{e}\right)^2 . \tag{19}$$

□

B. Lower bounds

The lower bounds of Section 4 rely on three steps: 1) defining a class of objective functions $\mathcal{L}^g$ for $g : \mathbb{R}^s \to \{-1, 1\}$ such that their optimum over $\mathcal{D}$ does not provide any information on the dataset $\mathcal{D}_r$, 2) showing that two such functions $\mathcal{L}^g$ and $\mathcal{L}^{-g}$ have optimums over $\mathcal{D}_r$ distant from one another, and 3) showing that any algorithm's output will behave nearly identically on both $\mathcal{L}^g$ and $\mathcal{L}^{-g}$, thus leading to the impossibility of having both functions efficiently optimized by the same algorithm.

In what follows, for any function $g : \mathbb{R}^s \to [-1, 1]$, we denote as $\mathcal{L}^g(\boldsymbol{\theta}) = \mathbb{E}\left[\ell^g(\boldsymbol{\theta}, \xi)\right]$ where ℓ^g is a loss function such that

$$\ell^g(\boldsymbol{\theta}, \xi) = \frac{\mu}{2}\|\boldsymbol{\theta}\|^2 - \frac{L}{2} g(\xi)\boldsymbol{\theta}_1 , \tag{20}$$

where $\boldsymbol{\theta}_1$ is the first coordinate of $\boldsymbol{\theta}$ in the canonical basis of $\mathbb{R}^d$. By definition, $\nabla_{\boldsymbol{\theta}} \ell^g(\boldsymbol{\theta}, \xi) = \mu\boldsymbol{\theta} - Lg(\xi)e_1/2$ where e_1 is the first vector of the canonical basis of $\mathbb{R}^d$, and ℓ^g is L-Lipschitz and μ-strongly convex. Moreover, the objective function on $\mathcal{D}_r$ is $\mathcal{L}_r^g(\boldsymbol{\theta}) = \frac{\mu}{2}\|\boldsymbol{\theta}\|^2 - \frac{L}{2}\mathbb{E}\left[g(\xi')\right]\boldsymbol{\theta}_1$, where $\xi' \sim \mathcal{D}_r$, and thus the minimizer of $\mathcal{L}_r^g$ is $\boldsymbol{\theta}_{g,r}^* = \frac{L}{2\mu}\mathbb{E}\left[g(\xi')\right] e_1$ (note that $\|\boldsymbol{\theta}_{g,r}^*\| = \frac{L}{2\mu}|\mathbb{E}\left[g(\xi')\right]| \leq R$). We now show that, provided we find two functions g, g' such that the output of any algorithm is (statistically) almost indistinguishable, then minimizing both $\mathcal{L}^g$ and $\mathcal{L}^{g'}$ beyond a certain quantity is impossible. To properly define this *indistinguishability*, we will use the *total variation* distance $d_{\text{TV}}(P, Q) = \sup_{A \subset \mathbb{R}^s} |P(A) - Q(A)|$ for two probability distributions P and Q.

Lemma B.1. *Let $g, g' : \mathbb{R}^s \to [-1, 1]$ two functions, $\theta_0, \theta_0' \in \mathbb{R}^d$ two initial parameters, and $A \in \boldsymbol{A}$ an algorithm. Then*

$$\sup_{g'' \in \{g, g'\}} \mathbb{E}\left[\mathcal{L}_r^{g''}(\boldsymbol{\theta}_T^A(\boldsymbol{\theta}_0, \ell^{g''}, \mathcal{D}_r)) - \mathcal{L}_r^{g''*}\right] \geq \frac{L^2(\mathbb{E}\left[g(\xi')\right] - \mathbb{E}\left[g'(\xi')\right])^2}{32\,\mu}(1 - d_{\text{TV}}(P_g, P_{g'})) , \tag{21}$$

where $\xi' \sim \mathcal{D}_r$, $\mathcal{L}_r^{g''*} = \min_{\boldsymbol{\theta} \in \mathbb{R}^d} \mathcal{L}_r^{g''}(\boldsymbol{\theta})$ and P_g (resp. $P_{g'}$) is the probability distribution of $\boldsymbol{\theta}_T^A(\boldsymbol{\theta}_0, \ell^g, \mathcal{D}_r)$ (resp. $\boldsymbol{\theta}_T^A(\boldsymbol{\theta}_0', \ell^{g'}, \mathcal{D}_r)$).

Proof. First, note that $\boldsymbol{\theta}_{g,r}^* = \frac{L}{2\mu}\mathbb{E}\left[g(\xi')\right] e_1$ and thus $\mathcal{L}_r^g(\boldsymbol{\theta}) - \mathcal{L}_r^{g^*} = \frac{\mu}{2}\|\boldsymbol{\theta}_{g,r}^* - \boldsymbol{\theta}\|^2$. Using the optimal transport definition of total variation (see *e.g.,* Villani et al. 2009), $d_{\text{TV}}(P, Q) = \inf_{(X,Y)} \mathbb{P}(X \neq Y)$ where the infimum is taken over all couplings of P and Q. As a consequence, there exists two random variables $\boldsymbol{\theta}_1 \sim P_g$ and $\boldsymbol{\theta}_2 \sim P_{g'}$, and such that $\mathbb{P}(\boldsymbol{\theta}_1 \neq \boldsymbol{\theta}_2) = d_{\text{TV}}(P_g, P_{g'})$, leading to

$$\begin{aligned}
\sup_{g'' \in \{g, g'\}} \mathbb{E}\left[\left\|\boldsymbol{\theta}_{g'',r}^* - \boldsymbol{\theta}_T^A(\boldsymbol{\theta}_0, \ell^{g''}, \mathcal{D}_r)\right\|^2\right] &= \max\left\{\mathbb{E}\left[\left\|\boldsymbol{\theta}_{g,r}^* - \boldsymbol{\theta}_1\right\|^2\right], \mathbb{E}\left[\left\|\boldsymbol{\theta}_{g',r}^* - \boldsymbol{\theta}_2\right\|^2\right]\right\} \\
&\geq (1 - d_{\text{TV}}(P_g, P_{g'})) \max\left\{\left\|\boldsymbol{\theta}_{g,r}^* - \tilde{\boldsymbol{\theta}}\right\|^2, \left\|\boldsymbol{\theta}_{g',r}^* - \tilde{\boldsymbol{\theta}}\right\|^2\right\} \\
&\geq (1 - d_{\text{TV}}(P_g, P_{g'})) \frac{\left\|\boldsymbol{\theta}_{g,r}^* - \boldsymbol{\theta}_{g',r}^*\right\|^2}{4} ,
\end{aligned} \tag{22}$$

where $\tilde{\boldsymbol{\theta}} = \mathbb{E}\left[\boldsymbol{\theta}_1 \mid \boldsymbol{\theta}_1 = \boldsymbol{\theta}_2\right] = \mathbb{E}\left[\boldsymbol{\theta}_2 \mid \boldsymbol{\theta}_1 = \boldsymbol{\theta}_2\right]$. Finally, using the formula for $\boldsymbol{\theta}_{g,r}^* = \frac{L}{2\mu}\mathbb{E}\left[g(\xi')\right] e_1$ and $\mathcal{L}_r^g(\boldsymbol{\theta}) - \mathcal{L}_r^{g^*} = \frac{\mu}{2}\|\boldsymbol{\theta}_{g,r}^* - \boldsymbol{\theta}\|^2$ gives the desired result. □

We now show that a particular choice of functions g, g' leads to almost indistinguishable outputs.

Lemma B.2. *Assume that $\forall \gamma \in [0, 1]$, there exists $A^\gamma \subset supp(\mathcal{D}_r)$ such that $\mathbb{P}_{\mathcal{D}_r}(A) = (1 + \gamma)/2$. Let $g^\gamma(\xi) = 2\mathbf{1}\{\xi \in A^\gamma\} - 1$ if $\xi \in supp(\mathcal{D}_r)$, and $g^\gamma(\xi) = -\min\{1, \frac{(1-r_f)\gamma}{r_f}\}$ otherwise. Then, we have*

$$d_{\text{TV}}\left(\mathcal{U}(T, \ell^{g^\gamma}, \mathcal{D}_r, \mathcal{D}_f), \mathcal{U}(T, \ell^{g^0}, \mathcal{D}_r, \mathcal{D}_f)\right) \leq \frac{\pi\gamma\sqrt{T}}{4} + \frac{((1 - r_f)\gamma - r_f)_+}{2r_f}(e^\epsilon - 1 + \delta) . \tag{23}$$

Proof. First, note that the minimizer of $\mathcal{L}^\gamma$ is $\boldsymbol{\theta}^*_\gamma = \frac{L}{2\mu}((1-r_f)\gamma - r_f)_+ e_1$, and we thus have $\mathcal{U}(T, \ell^{g^\gamma}, \mathcal{D}_r, \mathcal{D}_f) = \boldsymbol{\theta}^A_T(\boldsymbol{\theta}^*_\gamma, \ell^{g^\gamma}, \mathcal{D}_r)$ and $\mathcal{U}(T, \ell^{-g^\gamma}, \mathcal{D}_r, \mathcal{D}_f) = \boldsymbol{\theta}^A_T(-\boldsymbol{\theta}^*_\gamma, \ell^{-g^\gamma}, \mathcal{D}_r)$. To ease the notations, we denote by $\boldsymbol{\theta}_{k,l} = \boldsymbol{\theta}^A_T(\frac{(1-r_f)L\gamma_{2k+1}}{2\mu} e_1, \ell^{g^{\gamma_{2l}}}, \mathcal{D}_r)$ the output of algorithm A on the function $\ell^{g^{\gamma_{2l}}}$ starting at $\boldsymbol{\theta}_0 = \frac{(1-r_f)L\gamma_{2k+1}}{2\mu} e_1$, where $\gamma_k = \left(\gamma - \frac{kr_f}{1-r_f}\right)_+$. Let $K = \left\lceil \frac{(1-r_f)\gamma}{2r_f} \right\rceil$, then we have, by triangular inequality,

$$d_{\text{TV}}(\boldsymbol{\theta}_{0,0}, \boldsymbol{\theta}_{K,K}) \leq \sum_{k=0}^{K-1} d_{\text{TV}}(\boldsymbol{\theta}_{k,k}, \boldsymbol{\theta}_{k,k+1}) + \sum_{k=0}^{K-1} d_{\text{TV}}(\boldsymbol{\theta}_{k,k+1}, \boldsymbol{\theta}_{k+1,k+1}) \,. \tag{24}$$

By construction, we have $\boldsymbol{\theta}_{0,0} = \mathcal{U}(T, \ell^{g^\gamma}, \mathcal{D}_r, \mathcal{D}_f)$ and $\boldsymbol{\theta}_{K,K} = \mathcal{U}(T, \ell^{g^0}, \mathcal{D}_r, \mathcal{D}_f)$. We now show that both sums can be bounded: the first using the fact that the T gradients $g^\gamma(\xi_t)$ for $t \in [\![0, T-1]\!]$ are close in total variation distance (*i.e.,* Lemma C.5), and the second using the (ϵ, δ)-Unlearning constraint on $\mathcal{U}$.

By Lemma C.4, there is a measurable function φ_A such that $\boldsymbol{\theta}_{k,l} = \varphi_A\left(\frac{(1-r_f)L\gamma_{2k+1}}{2\mu}, Z^l_0, \ldots, Z^l_{T-1}, \omega\right)$ where $Z^l_t = (1 + g^{\gamma_{2l}}(\xi_t))/2$ are i.i.d. Bernoulli random variables of parameter $\frac{1+\gamma_{2l}}{2}$. As $\boldsymbol{\theta}_{k,k}$ and $\boldsymbol{\theta}_{k,k+1}$ are outputs of the same algorithm initialized at the same starting position, we have

$$\begin{aligned}
\textstyle\sum_{k=0}^{K-1} d_{\text{TV}}(\boldsymbol{\theta}_{k,k}, \boldsymbol{\theta}_{k,k+1}) &= \textstyle\sum_{k=0}^{K-1} d_{\text{TV}}\left((Z^k_0, \ldots, Z^k_{T-1}), (Z^{k+1}_0, \ldots, Z^{k+1}_{T-1})\right) \\
&= d_{\text{TV}}\left(\text{Bin}\left(T, \tfrac{1+\gamma_{2k}}{2}\right), \text{Bin}\left(T, \tfrac{1+\gamma_{2k+2}}{2}\right)\right) \\
&= \textstyle\sum_{k=0}^{K-1} \frac{\sqrt{T}}{2} \left| \tan^{-1}\left(\frac{\gamma_{2k+2}}{\sqrt{1-\gamma_{2k+2}{}^2}}\right) - \tan^{-1}\left(\frac{\gamma_{2k}}{\sqrt{1-\gamma_{2k}^2}}\right) \right| \\
&= \frac{\sqrt{T}}{2} \left| \tan^{-1}\left(\frac{\gamma_{2K}}{\sqrt{1-\gamma_{2K}{}^2}}\right) - \tan^{-1}\left(\frac{\gamma_0}{\sqrt{1-\gamma_0^2}}\right) \right| \\
&= \frac{\sqrt{T}}{2} \tan^{-1}\left(\frac{\gamma_{2K}}{\sqrt{1-\gamma_{2K}{}^2}}\right) \\
&\leq \frac{\pi\gamma\sqrt{T}}{4} \,,
\end{aligned} \tag{25}$$

using the fact that $f : x \mapsto \tan^{-1}\left(\frac{x}{\sqrt{1-x^2}}\right)$ is increasing and convex on $x \in [0,1]$, and $f(0) = 0$ and $f(1) = \pi/2$.

Finally, let $\mathcal{D}'_f$ be a probability distribution on $\mathbb{R}^s$ such that $\text{supp}(\mathcal{D}'_f) \cap (\text{supp}(\mathcal{D}_r) \cup \text{supp}(\mathcal{D}_f)) = \emptyset$, for any $\gamma \in [0,1]$, let $\tilde{g}^\gamma(\xi) = g^\gamma(\xi)$ if $\xi \in \text{supp}(\mathcal{D}_r) \cup \text{supp}(\mathcal{D}_f)$, and $\tilde{g}^\gamma(\xi) = 1$ otherwise. Then, we have $\mathcal{U}(T, \ell^{\tilde{g}^{\gamma_{2k}}}, \mathcal{D}_r, \mathcal{D}_f) = \boldsymbol{\theta}_{k,k}$ and $\mathcal{U}(T, \ell^{\tilde{g}^{\gamma_{2k}}}, \mathcal{D}_r, \mathcal{D}'_f) = \boldsymbol{\theta}_{k-1,k}$. Thus, we have

$$\begin{aligned}
\textstyle\sum_{k=0}^{K-1} d_{\text{TV}}(\boldsymbol{\theta}_{k,k+1}, \boldsymbol{\theta}_{k+1,k+1}) &= \textstyle\sum_{k=0}^{K'-1} d_{\text{TV}}\left(\mathcal{U}(T, \ell^{\tilde{g}^{\gamma_{2k+2}}}, \mathcal{D}_r, \mathcal{D}_f), \mathcal{U}(T, \ell^{\tilde{g}^{\gamma_{2k+2}}}, \mathcal{D}_r, \mathcal{D}'_f)\right) \\
&\leq \textstyle\sum_{k=0}^{K'-1} (e^\epsilon - 1 + \delta) \\
&= K'(e^\epsilon - 1 + \delta) \,,
\end{aligned} \tag{26}$$

where $K' = \left\lceil \frac{((1-r_f)\gamma - r_f)_+}{2r_f} \right\rceil$. Combining the two inequalities concludes the proof. □

We are now in position to prove Theorem 2.

Proof of Theorem 2. Combining Lemma B.1 (with $g = g^\gamma$ and $g' = -g^\gamma$) and Lemma B.2, we have, for any $\gamma \in [0,1]$,

$$\min_{\mathcal{U} \in \mathbb{U}_{\epsilon,\delta}} \max_{\mathcal{L} \in \mathcal{F}_{sc}} \mathbb{E}\left[\mathcal{L}_r(\mathcal{U}(T, \ell, \mathcal{D}_r, \mathcal{D}_f)) - \mathcal{L}^*_r\right] \geq \frac{L^2\gamma^2}{8\,\mu}(1 - d_{\text{TV}}(P_{g^\gamma}, P_{-g^\gamma})) \,, \tag{27}$$

where $d_{\text{TV}}(P_{g^\gamma}, P_{-g^\gamma}) \leq d_{\text{TV}}(P_{g^\gamma}, P_{g^0}) + d_{\text{TV}}(P_{g^0}, P_{-g^\gamma}) \leq \frac{\pi\gamma\sqrt{T}}{2} + \frac{((1-r_f)\gamma - r_f)_+}{r_f}(e^\epsilon - 1 + \delta)$. Let $c_1, c_2 \in [0,1]$ and $\gamma = c_1/\sqrt{T}$. If $\left(\frac{(1-r_f)c_1}{r_f\sqrt{T}} - 1\right)_+ (e^\epsilon - 1 + \delta) \leq c_2$, then

$$\min_{\mathcal{U} \in \mathbb{U}_{\epsilon,\delta}} \max_{\mathcal{L} \in \mathcal{F}_{sc}} \mathbb{E}\left[\mathcal{L}_r(\mathcal{U}(T, \ell, \mathcal{D}_r, \mathcal{D}_f)) - \mathcal{L}^*_r\right] \geq \frac{L^2 c_1^2}{8\,\mu T}(1 - \frac{\pi c_1}{2} - c_2) \,, \tag{28}$$

and thus, if $\left(\frac{(1-r_f)\sqrt{8\mu e}}{r_f L\sqrt{1-\frac{\pi c_1}{2}-c_2}}-1\right)_+(e^\epsilon-1+\delta)\leq c_2$,

$$T_e^U \geq \frac{L^2c_1^2}{8\,\mu e}(1-\frac{\pi c_1}{2}-c_2)\,. \tag{29}$$

Finally, we take $c_2=1/2$, c_1 such that $1-\frac{\pi c_1}{2}-c_2=1/3$, and rewrite the condition as $e\leq\frac{r_f^2L^2}{8\,(1-r_f)^2\mu}(1-\frac{\pi c_1}{2}-c_2)\left(1+\frac{c_2}{e^\epsilon-1+\delta}\right)^2$. A simple functional analysis gives that, for $10^{-8}\leq\delta\leq\epsilon$, we have

$$1+\frac{1}{2(e^\epsilon-1+\delta)}\geq 1+\frac{1}{2(e^\epsilon-1+\epsilon)}\geq c_3\left(1+\frac{\sqrt{2\ln(1.25\cdot 10^8)}}{\epsilon}\right)\geq c_3\left(1+\frac{\sqrt{2\ln(1.25/\delta)}}{\epsilon}\right)\,, \tag{30}$$

where $c_3=1/\sqrt{32\ln(1.25\cdot 10^8)}$ and the desired result. □

Using the same approach, a lower bound on the time complexity of scratch can also be derived.

Lemma B.3. *Under Assumption 2, and if $e<e_0$, we have*

$$T_e^S=\Theta\left(\frac{e_0}{e}\right)\,. \tag{31}$$

Proof of Lemma B.3. First, by strong convexity, we have

$$\mathcal{L}_r(0)-\mathcal{L}_r^*\leq\langle\nabla\mathcal{L}(0),\boldsymbol{\theta}_r^*\rangle-\frac{\mu}{2}\left\|\boldsymbol{\theta}_r^*\right\|^2\,. \tag{32}$$

Moreover, the convexity of $\boldsymbol{\theta}\mapsto\mathcal{L}_r(\boldsymbol{\theta})-\frac{\mu}{2}\left\|\boldsymbol{\theta}\right\|^2$ implies that $\langle\nabla\mathcal{L}(-\boldsymbol{\theta}_r^*)+\mu\boldsymbol{\theta}_r^*-\nabla\mathcal{L}(0),-\boldsymbol{\theta}_r^*\rangle\geq 0$ and thus

$$\mathcal{L}_r(0)-\mathcal{L}_r^*\leq\left\|\nabla\mathcal{L}(-\boldsymbol{\theta}_r^*)\right\|\left\|\boldsymbol{\theta}_r^*\right\|-\frac{3\mu}{2}\left\|\boldsymbol{\theta}_r^*\right\|^2\leq LR-\frac{3\mu}{2}R^2=\frac{L^2}{8\mu}\,. \tag{33}$$

As a consequence, if $e\geq e_0$, then $T_e^S=0$ (and $T_e^U=0$). Let us now assume that $e<e_0$. First, note that this convergence rate is achieved by stochastic gradient descent. For example, a direct extension of Theorem 6.2 from Bubeck et al. (2015) gives, after T iterations of (stochastic) gradient descent $\theta_{t+1}=\theta_t-\eta_t\nabla_\theta\ell(\theta_t,\xi_t)$ with decreasing step-size $\eta_t=\frac{2}{\mu(t+2)}$.

$$\mathbb{E}\left[\mathcal{L}\left(\tilde{\theta}_T\right)-\min_{\theta\in\mathbb{R}^d}\mathcal{L}(\theta)\right]\leq\frac{2L^2}{\mu(T+2)}\,, \tag{34}$$

where $\tilde{\theta}_T=\sum_{t=0}^{T-1}\frac{2(1+t)}{(T+1)(T+2)}\theta_t$, and thus

$$T_e^S\leq\frac{2L^2}{\mu e}\,. \tag{35}$$

The lower bound is a consequence of Theorem 2 with $\kappa_{\epsilon,\delta}=0$, as an algorithm retraining from scratch would not depend on the forget dataset, and thus have absolute privacy. In particular, if $e\leq\frac{L^2}{8\,\mu}(1-\frac{\pi c_1}{2})$, we have

$$T_e^S\geq\frac{L^2c_1^2}{8\,\mu e}(1-\frac{\pi c_1}{2})\,, \tag{36}$$

and as soon as $e/e_0\leq 1-\eta$ for $\eta>0$, there exists a constant $c_2>0$ such that $T_e^S\geq c_2e_0/e$. □

C. Useful lemmas

In this section, we provide five lemmas that will be necessary to prove our upper and lower bounds (see sections above), as well as the proof for unlearning definition equivalence.

Lemma C.1. *Let $\boldsymbol{\theta}_r^* = \arg\min_{\boldsymbol{\theta}} \mathcal{L}_r(\boldsymbol{\theta})$. Then, we have:*

$$\|\boldsymbol{\theta}^* - \boldsymbol{\theta}_r^*\| \leq \frac{r_f}{1-r_f} \cdot \frac{L}{\mu}\,. \tag{37}$$

Proof. By strong convexity of $\mathcal{L}_r$, we have $\|\boldsymbol{\theta}^* - \boldsymbol{\theta}_r^*\| \leq \frac{\|\nabla \mathcal{L}_r(\boldsymbol{\theta}^*)\|}{\mu}$. Moreover, $\|\nabla \mathcal{L}_r(\boldsymbol{\theta}^*)\| = \|\mathbb{E}\left[\nabla \ell(\boldsymbol{\theta}^*, \xi_r)\right]\| = \left\|-\frac{r_f}{1-r_f}\,\mathbb{E}\left[\nabla \ell(\boldsymbol{\theta}^*, \xi_r)\right]\right\| \leq \frac{r_f}{1-r_f} L$ where $\xi_r \sim \mathcal{D}_r$ and $\xi_f \sim \mathcal{D}_f$, as $\nabla \mathcal{L}(\boldsymbol{\theta}^*) = 0$. Combining the two inequalities gives the desired result. □

Lemma C.2.

$$\mathcal{L}_r(\boldsymbol{\theta}^*) - \mathcal{L}_r^* \leq \left(\frac{r_f}{1-r_f}\right)^2 \frac{L^2}{\mu} \tag{38}$$

Proof. Let $\boldsymbol{\theta}_r^* = \arg\min_{\boldsymbol{\theta}} \mathcal{L}_r(\boldsymbol{\theta})$. Then,

$$\begin{aligned}
\mathbb{E}_{\mathcal{D}_r}\left(\ell(\boldsymbol{\theta}^*) - \ell(\boldsymbol{\theta}_r^*)\right) &= \mathbb{E}_{\mathcal{D}}\left(\ell(\boldsymbol{\theta}^*) - \ell(\boldsymbol{\theta}_r^*)\right) - \frac{r_f}{1-r_f}\mathbb{E}_{\mathcal{D}_f}\left(\ell(\boldsymbol{\theta}^*) - \ell(\boldsymbol{\theta}_r^*)\right) && (39)\\
&\leq -\frac{r_f}{1-r_f}\mathbb{E}_{\mathcal{D}_f}\left(\ell(\boldsymbol{\theta}^*) - \ell(\boldsymbol{\theta}_r^*)\right) && (40)\\
&\leq \frac{r_f}{1-r_f} L\,\|\boldsymbol{\theta}^* - \boldsymbol{\theta}_r^*\| && (41)\\
&\leq \left(\frac{r_f}{1-r_f}\right)^2 \frac{L^2}{\mu}, && (42)
\end{aligned}$$

where the last inequality is given by Lemma C.1. □

Lemma C.3. *If the unlearning algorithm $\mathcal{U}$ verifies (ϵ, δ)-Unlearning, then, for any triplet of distributions $(\mathcal{D}_r, \mathcal{D}_f, \mathcal{D}_f')$, we have*

$$d_{\mathrm{TV}}\left(\mathcal{U}(\mathcal{D}_r, \mathcal{D}_f), \mathcal{U}(\mathcal{D}_r, \mathcal{D}_f')\right) \leq e^{\epsilon} - 1 + \delta\,. \tag{43}$$

Proof. By (ϵ, δ)-Unlearning, we have, for any $S \subset \mathbb{R}^s$, $\mathbb{P}[\mathcal{U}(\mathcal{D}_r, \mathcal{D}_f) \in S] \leq e^{\epsilon} \cdot \mathbb{P}[\mathcal{U}(\mathcal{D}_r, \mathcal{D}_f') \in S] + \delta$, and thus

$$\mathbb{P}[\mathcal{U}(\mathcal{D}_r, \mathcal{D}_f) \in S] - \mathbb{P}[\mathcal{U}(\mathcal{D}_r, \mathcal{D}_f') \in S] \leq (e^{\epsilon} - 1) \cdot \mathbb{P}[\mathcal{U}(\mathcal{D}_r, \mathcal{D}_f') \in S] + \delta \leq e^{\epsilon} - 1 + \delta\,. \tag{44}$$

The converse relation with $\mathcal{D}_f$ and $\mathcal{D}_f'$ exchanged leads to a bound on the absolute value, and thus the desired result. □

Proof of Lemma **??**. Let $\mathcal{D}_0$ be an arbitrary distribution, *e.g.,* the uniform distribution on the $R/2$-ball. Let $\mathcal{U} \in \mathbb{U}_{\epsilon,\delta}$ be an (ϵ, δ)-Unlearning algorithm. Let $\mathcal{U} \in \mathbb{U}_{\epsilon,\delta}$ be an (ϵ, δ)-Unlearning algorithm. Then, the algorithm $\mathcal{A}_0 : (T, l, \mathcal{D}_r) \longmapsto \mathcal{U}(T, l, \mathcal{D}_r, \mathcal{D}_0)$ is such that for any couple of distributions $(\mathcal{D}_r, \mathcal{D}_f)$ over $\mathbb{B}(0, R)$ and subset $S \subset \mathbb{R}^d$,

$$\begin{aligned}
\mathbb{P}[\mathcal{U}(\mathcal{D}_r, \mathcal{D}_f) \in S] &\leq e^{\epsilon} \cdot \mathbb{P}[\mathcal{U}(\mathcal{D}_r, \mathcal{D}_0) \in S] + \delta = e^{\epsilon} \cdot \mathbb{P}[\mathcal{A}_0(\mathcal{D}_r) \in S] + \delta,\\
\mathbb{P}[\mathcal{A}_0(\mathcal{D}_r) \in S] = \mathbb{P}[\mathcal{U}(\mathcal{D}_r, \mathcal{D}_0) \in S] &\leq e^{\epsilon} \cdot \mathbb{P}[\mathcal{U}(\mathcal{D}_r, \mathcal{D}_f) \in S] + \delta.
\end{aligned}$$

$\mathcal{U}$ is thus an (ϵ, δ)-Reference Unlearning algorithm. This proves the first implication.

Let $\mathcal{U} \in \mathbb{U}_{\epsilon,\delta}$ be an (ϵ, δ)-Reference Unlearning algorithm and $\mathcal{A} \in \mathbb{A}$ its reference algorithm. Let $\mathcal{D}_r, \mathcal{D}_f, \mathcal{D}_f'$ be three distributions over $\mathbb{B}(0, R)$. Then,

$$\mathbb{P}[\mathcal{U}(\mathcal{D}_r, \mathcal{D}_f) \in S] \leq e^{\epsilon} \cdot \mathbb{P}[\mathcal{A}(\mathcal{D}_r) \in S] + \delta \leq e^{\epsilon}\left(e^{\epsilon} \cdot \mathbb{P}[\mathcal{U}(\mathcal{D}_r, \mathcal{D}_f') \in S] + \delta\right) + \delta \tag{45}$$

$\mathcal{U}$ is thus an $(2\epsilon, (1 + \exp(\epsilon))\delta)$-Reference Unlearning algorithm. This concludes the proof. □

Lemma C.4. *Let $A \in \boldsymbol{A}$ be an iterative algorithm as defined in Algorithm 1. Then, there exists T i.i.d. random variables $\xi_t \sim \mathcal{D}_r$ and a measurable function $\varphi_{\mathcal{A}}$ such that, for any $T > 0$, $\theta_0 \in \mathbb{R}^d$ and $g : \mathbb{R}^s \to \{-1, 1\}$, we have*

$$\boldsymbol{\theta}_T^A(\boldsymbol{\theta}_0, \ell^g, \mathcal{D}_r) = \varphi_A(\boldsymbol{\theta}_0, g(\xi_0), \ldots, g(\xi_{T-1}), \omega)\,. \tag{46}$$

Proof. Let $(m_t, \boldsymbol{\theta}_t)$ be the memory state and current parameter of algorithm A at iteration t (see algorithm 1). First, for $T = 0$, $m_0 = \emptyset$ and $\boldsymbol{\theta}_0$ are both (trivially) measurable functions of $\boldsymbol{\theta}_0$. Then, by recursion, if both m_{T-1} and $\boldsymbol{\theta}_{T-1}$ are measurable functions of $\boldsymbol{\theta}_0, g(\xi_0), \dots, g(\xi_{T-2}), \omega$, then

$$(m_T, \boldsymbol{\theta}_T) = A(m_t, \nabla \ell^g(\boldsymbol{\theta}_{T-1}, \xi_{T-1}), \omega) = A(m_{T-1},\, \mu\boldsymbol{\theta}_{T-1} - Lg(\xi_{T-1})e_1/2,\, \omega)\,, \tag{47}$$

which is a measurable function of $\boldsymbol{\theta}_0, g(\xi_0), \dots, g(\xi_{T-1}), \omega$. This concludes the proof. □

Lemma C.5. *Let $T \geq 0$ and $\gamma, \gamma' \in [-1, 1]$. Then, we have*

$$d_{\text{TV}}\left(\textit{Bin}\left(T, \frac{1+\gamma}{2}\right), \textit{Bin}\left(T, \frac{1+\gamma'}{2}\right)\right) \leq \frac{\sqrt{T}}{2}\left|\tan^{-1}\left(\frac{\gamma'}{\sqrt{1-\gamma'^2}}\right) - \tan^{-1}\left(\frac{\gamma}{\sqrt{1-\gamma^2}}\right)\right|. \tag{48}$$

Proof. Assume that $\gamma' \geq \gamma$, and let $\varphi(\gamma, \gamma') = d_{\text{TV}}\left(\text{Bin}\left(T, \frac{1+\gamma}{2}\right), \text{Bin}\left(T, \frac{1+\gamma'}{2}\right)\right)$. The proof relies on bounding the derivative of φ with respect to its second variable. Let $\gamma' = \gamma + \varepsilon$ where $\varepsilon > 0$, then

$$\begin{aligned}
\varphi(\gamma, \gamma+\varepsilon) &= \tfrac{1}{2}\,\mathbb{E}\left[\left|1 - \tfrac{P_{\gamma+\varepsilon}(X)}{P_{\gamma(X)}}\right|\right] \\
&= \tfrac{1}{2}\,\mathbb{E}\left[\left|1 - \tfrac{(1+\gamma+\varepsilon)^X(1-\gamma-\varepsilon)^{T-X}}{(1+\gamma)^X(1-\gamma)^{T-X}}\right|\right] \\
&= \tfrac{1}{2}\,\mathbb{E}\left[\left|1 - (1 + X\tfrac{\varepsilon}{1+\gamma})(1 - (T-X)\tfrac{\varepsilon}{1-\gamma}) + O(\varepsilon^2)\right|\right] \\
&= \tfrac{\varepsilon}{2(1+\gamma)}\,\mathbb{E}\left[\left|X - (T-X)\tfrac{1+\gamma}{1-\gamma}\right|\right] + O(\varepsilon^2) \\
&\leq \tfrac{\varepsilon}{(1+\gamma)}\sqrt{\tfrac{\text{Var}(X)}{(1-\gamma)^2}} + O(\varepsilon^2) \\
&= \tfrac{\varepsilon}{2}\sqrt{\tfrac{T}{1-\gamma^2}} + O(\varepsilon^2)\,,
\end{aligned} \tag{49}$$

where P_γ is the density of the binomial distribution $\text{Bin}\left(T, \frac{1+\gamma}{2}\right)$, and $X \sim \text{Bin}\left(T, \frac{1+\gamma}{2}\right)$. As the total variation distance verifies the triangular inequality, we have

$$d_{\text{TV}}\left(\text{Bin}\left(T, \frac{1+\gamma}{2}\right), \text{Bin}\left(T, \frac{1+\gamma'}{2}\right)\right) \leq \int_{u=\gamma}^{\gamma'} \varphi(u, u{+}du) \leq \frac{\sqrt{T}}{2}\left(\tan^{-1}\left(\frac{\gamma'}{\sqrt{1-\gamma'^2}}\right) - \tan^{-1}\left(\frac{\gamma}{\sqrt{1-\gamma^2}}\right)\right). \tag{50}$$

□

D. Algorithms

Algorithm 1 Iterative (Un)Learning Algorithm

Require: Update rule $A \in \boldsymbol{A}$, number of iterations T, initial model $\boldsymbol{\theta}_0$, loss function ℓ, dataset $\mathcal{D}$.

```
Initialize memory: m_0 = ∅
for t = 0 to T − 1 do
    Sample data point: ξ_t ∼ D
    Compute gradient: ∇ℓ(θ_t, ξ_t)
    Update: (θ_{t+1}, m_{t+1}) = A(θ_t, ∇ℓ(θ_t, ξ_t), m_t, ω)
end for
return Final model θ_T
```

E. Iterative algorithms based on update rules

In this, section, we provide precise definitions for learning and unlearning algorithms. More precisely, we will consider that both types of algorithms are *non-deterministic*, *iterative* and *first-order*, *i.e.,* that model parameters are updated through a stochastic iterative procedure that accesses a stochastic gradient of the loss function at each iteration (see Algorithm 1). This

Algorithm 2 "Noise and Fine-Tune" Unlearning Algorithm

Require: number of iterations T, initial model $\boldsymbol{\theta}^*$, loss function ℓ, dataset $\mathcal{D}_r$.

Sample noise $g \sim \mathcal{N}\left(0, \left(\kappa_{\epsilon,\delta} r_f \frac{L}{\mu}\right)^2 I_d\right)$
Initialize model: $\boldsymbol{\theta}_0 = \boldsymbol{\theta}^* + g$
Initialize memory: $m_0 = \boldsymbol{\theta}_0$
for $t = 1$ to T **do**
  Sample data point: $\xi_t \sim \mathcal{D}_r$
  Compute gradient: $\nabla \ell(\boldsymbol{\theta}_t, \xi_t)$
  Update: $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \frac{2}{\mu(t+1)} \nabla \ell(\boldsymbol{\theta}_t, \xi_t)$
  Update: $m_{t+1} = m_t + (t+1)\boldsymbol{\theta}_{t+1}$
end for
return Final model $\hat{\boldsymbol{\theta}} = \frac{2m_T}{(T+1)(T+2)}$

class of algorithms, defined by their *update rule* $A \in \boldsymbol{A}$, is very general and contains most standard optimization algorithms used in machine learning. More precisely, an update rule is a measurable function

$$A\left(\boldsymbol{\theta}_t, \nabla_t, m_t, \omega\right) = \left(\boldsymbol{\theta}_{t+1}, m_{t+1}\right), \tag{51}$$

where $\boldsymbol{\theta}_t \in \mathbb{R}^d$ is the current model, $\nabla_t \in \mathbb{R}^d$ a stochastic gradient, $m_t \in M$ a memory state, and $\omega \in \Omega$ a seed used for adding randomness into the algorithm. The memory serves as a storage mechanism for essential information about past iterates, enabling the computation of quantities such as momentum, moving averages, or adaptive step-sizes.

For a given update rule $A \in \boldsymbol{A}$, we denote as $\boldsymbol{\theta}_T^A(\boldsymbol{\theta}_0, \ell, \mathcal{D}_r)$ the output of Algorithm 1, which applies the update rule A successively T times, starting at $\boldsymbol{\theta}_0 \in \mathbb{R}^d$.

Learning Algorithms. For any update rule $A \in \boldsymbol{A}$, we define the associated *learning algorithm* as the function $\mathcal{A}$ mapping the number of iterations, loss function, and dataset to the output of A initialized at $\boldsymbol{\theta}_0 = 0$, *i.e.,*

$$\mathcal{A}(T, \ell, \mathcal{D}_r) = \boldsymbol{\theta}_T^A(0, \ell, \mathcal{D}_r)\,. \tag{52}$$

In the rest of the paper, we denote by $\mathbb{A}$ the class of such learning algorithms, and write $\mathcal{A}(\mathcal{D}_r)$ when there is no ambiguity on the values of T and ℓ.

Unlearning Algorithms. While learning algorithms try to estimate the optimum of the objective function $\mathcal{L}_r$ from scratch, unlearning algorithms have the advantage of starting from a pre-trained model with low excess risk (*i.e.,* error of the model minus error of the optimal model) on the whole dataset. More precisely, we will assume that such model was trained for a sufficiently large amount of time, and reached the unique minimizer $\boldsymbol{\theta}^*$ of the objective function $\mathcal{L}$. Therefore, for any update rule $A \in \boldsymbol{A}$, we define the associated *unlearning algorithm* as the function $\mathcal{U}$ mapping the number of iterations, loss function, retain dataset and forget dataset to the output of A initialized at $\boldsymbol{\theta}_0 = \boldsymbol{\theta}^*$, *i.e.,*

$$\mathcal{U}(T, \ell, \mathcal{D}_r, \mathcal{D}_f) = \boldsymbol{\theta}_T^A(\boldsymbol{\theta}^*, \ell, \mathcal{D}_r)\,. \tag{53}$$

Again, we will denote as $\mathbb{U}$ the class of such unlearning algorithms, and simply write $\mathcal{U}(\mathcal{D}_r, \mathcal{D}_f)$ when there is no ambiguity on the values of T and ℓ. Note that these unlearning algorithms can only sample from the retain set to perform unlearning. This is relatively common in the literature of DP-based MU (Neel et al., 2021; Fraboni et al., 2024; Huang and Canonne, 2023; Allouah et al., 2024), although more efficient unlearning methods might exist in scenarios in which the forget dataset is also available during unlearning. Finally, while we allow stateful algorithms in our framework, the algorithm used to achieve our upper bound in Section 4 only uses the state to remember the weighted average of previous iterations rather than all iterations, alleviating some privacy issues for adaptive unlearning requests (Izzo et al., 2021).