BEHAVIORAL GUARANTEES FOR PROXY BASED UNLEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper generalizes recent proxy based unlearning methods and proves theoretical guarantees about the behavior of the resulting unlearned model: upper bounds on the Kullback-Leibler divergence from the ideal retrained. We model approximate unlearning as a constrained optimization problem and interpret a family of solutions as introducing a scaled unlearning signal in the output space. The unlearning signal arises from proxies of the posterior data distributions. Its scale is adapted to the proxies to ensure the behavioral upper bounds. This framework relies on the structure of the data distributions in order to create proxies. If need be, the target serves as a teacher to distill the update in the weights. Our approach is experimentally validated over two forgetting scenarios as reaching the closest classifier to the ideal retrained model.

1 INTRODUCTION

Approximate machine unlearning removes the influence of specific training data from an already trained classifier. This avoids retraining it from scratch. Applications range from privacy (the right to be forgotten) to safety/security by removing corrupted/poisoned training data. Another scenario is deriving a restricted public model from a powerful private model trained on sensitive data.

Classical gradient-based methods construct asymmetrical losses that treat retain or forget samples differently. They are iterative, hyperparameter-sensitive, and lack a stopping criterion. This leads to unstable behaviors. Their efficacy is often gauged by misleading accuracy metric interpretations or oracles, without accounting for the ideal retrained model.

This paper first defines the unlearning signal as the logits shift between the ideal and the initial output distributions. Our method estimates this shift in the output space using posteriors of a pair of proxies. Section 3 bounds the gap to the ideal retrained model by an initial divergence plus a modeling shift. It proves that the bound is small due to the compensation between the two proxies errors, provided that the scale of the unlearning signal remains under a safe value $\eta_{\max}$. This unlearning signal is added on top of a black-box model, or cast within the neural network parameters via distillation. With the teacher fixed, this final step is fast, stable, and needs no stopping criterion.

A key element of our method is the choice of the proxies. We construct them by explicitly modeling the data distributions rather than by learning posteriors. Section 4 proposes a family of proxies ranging from naive marginalization to more advanced modeling specifically designed for unlearning.

Our contributions are the following. *(i)* Theoretical guarantees on the behavior of the unlearned model (Sect. 3). *(ii)* Practical implementation with interpretable unlearning signals stemming from data modeling proxies (Sect. 4) and where the safe scale $\eta_{\max}$ is derived by a simple 1-d line search. *(iii)* An experimental validation across four architectures, including a foundation model backbone, and two scenarios of unlearning demonstrating the versatility of the proposed method (Sect. 5).

2 RELATED WORK

Closed-form Modeling. Class-conditional densities have a rich history in statistics. Gaussian discriminant analysis (LDA, QDA) provides the canonical example, with regularized variants (Friedman, 1989) and Bayesian counterparts (Murphy, 2012; Gelman et al., 2013). More flexible options include

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.

per-class Gaussian mixtures (Dempster et al., 1977; McLachlan and Peel, 2000) and constrained covariance structures (Ghahramani and Hinton, 1996). Our method leverages these fast closed-form approaches. To our knowledge, no prior work has exploited the closed-form modeling by lifting the label space in a higher dimension to model the membership as a subclass, as proposed in Sect. 4.

Machine unlearning. It is the art of removing the influence of some training data without learning a new model from scratch (see App. A for a deeper overview). The gold standard is exact removal by design. Exact techniques rely on specific architectures or data structures to avoid full retraining (Cao and Yang, 2015). For instance, Bourtoule et al. (2021) propose the method SISA, which splits the training data into shards and stores intermediate checkpoints for each shard. The trade-off between memory footprint and utility is the main limitation at scale, especially when the data to be forgotten spans multiple shards.

Approximate unlearning. It aims at making the unlearned model statistically indistinguishable from the ideal model that never saw the forget data. The dominant paradigm, so-called first-order methods, consists of gradient-based updates of the NN parameters. Gradient ascent on the forget-set loss is the simplest primitive, many methods instead change the labels (Graves et al., 2021; Chen et al., 2023). More surgical method like SalUn (Fan et al., 2024) restricts updates to the weights most responsible for forgotten data. Distillation variants enforce retention jointly with forgetting. For instance, SCRUB trains a student by minimizing KL divergence with the original model on the data to retain while maximizing it on the data to forget (Kurmanji et al., 2023). Chundawat et al. (2023) use an incompetent teacher guiding the student to suppress class knowledge. Tarun et al. (2023) propose an impair-repair scheme: an error-maximizing noise degrades class knowledge, then a repair step restores retain-set accuracy. These first-order methods are unified into either a multi-objective or a constrained optimization (Zhou et al., 2025; Dine et al., 2026): unlearning the forget data while preserving the model utility. These papers outline the conflicting gradients of the objectives and propose ad-hoc combinations. Yet, the iterative updates produce a trajectory with no clear stopping criterion. Moreover, Mavrothalassitis et al. (2025) showed that dependence between forget and retain data prevents these first-order methods from unlearning properly.

Our formulation of the unlearning (1) is also framed as a constrained optimization problem. Yet, it has a closed-form solution which offers a theoretically grounded model correction, so-called unlearning signal. Distillation into the model is only a final option where the unlearning signal acts as a fixed teacher, so no catastrophic stopping criterion is needed.

Parametric models are rare in the machine unlearning literature. Georgiev et al. (2025) propose linear data model matching as a way to predict the output of the retrained model. We show that approximating only the retain set is sub-optimal. Section 4 builds pairs of proxies per class by lifting the label to (y, s), where y is a class and s indicates membership of the forget set.

From the audit point of view, the KL divergence from the ideal model to the empirically unlearned NN has been adopted as a natural criterion (Golatkar et al., 2020a; Chien et al., 2024; Georgiev et al., 2025; Ribero et al., 2026). Yet, nearly all theoretical results are stated in terms of NN properties rather than of data distribution. Indeed, the NeurIPS unlearning competition Triantafillou et al. (2024) relies on (ϵ, δ) certification (Sekhari et al., 2021) or membership inference attacks (MIAs) (Shokri et al., 2017; Carlini et al., 2022; Hayes et al., 2025) to evaluate unlearning. Those quantities are accessible to the auditor, but they cannot be used as guidance for the unlearner.

As for LLMs, recent approaches add a logits correction as we do. ULD (Ji et al., 2024) trains a small assistant overfitting the forget set and subtracts its logits from the initial model. UCD (Suriyakumar et al., 2025) uses contrastive approach with two auxiliary models. T3-Unlearning (Block et al., 2026) tilts and raises the temperature. In all three, their scaling is a free hyper-parameter tuned by hand. Our framework provides theoretical grounding (see App. G) and lets us derive a sound scaling $\eta_{\max}$. The propositions of Sect. 3 do apply to these three methods, but they are not evaluated here as this paper focuses on classifiers.

3 Behavioral guarantees of proxy modeling

Notations (summarized in Table 3, App B). The original training set $\mathcal{D} = \mathcal{D}_r \cup \mathcal{D}_f$ is composed by the forget set $\mathcal{D}_f$ to be erased and the retain set $\mathcal{D}_r$ to be preserved. Given a neural network (NN : functions from the input space $\mathbb{X}$ to the *logit* space $\mathbb{R}^C$ with C classes) $\boldsymbol{\ell_\theta}$ trained on $\mathcal{D}$, machine

unlearning seeks an efficient procedure producing NN $\boldsymbol{\ell}_{\boldsymbol{\theta}_u}$ as if it had been trained on $\mathcal{D}_r$ alone. The softmax operator yields predicted class probabilities a.k.a. *probits* $f_{\boldsymbol{\theta}}(\boldsymbol{x}) \doteq \mathrm{softmax}(\boldsymbol{\ell}_{\boldsymbol{\theta}}(\boldsymbol{x}))$. Conversely, $\boldsymbol{\ell}_{\boldsymbol{\theta}}(\boldsymbol{x}) = \log(f_{\boldsymbol{\theta}}(\boldsymbol{x})) + \gamma(\boldsymbol{x})\mathbf{1}$ for some $\gamma(\boldsymbol{x}) \in \mathbb{R}$. We denote by p the initial joint distribution of the data $(y, \boldsymbol{x})$. The induced marginal and conditioned true distributions are denoted $p_{\boldsymbol{x}}$, p_y and $p\left(y \mid \boldsymbol{x}\right)$, $p\left(\boldsymbol{x} \mid y\right)$, or simply p when the context is clear. The Kullback-Leibler divergence from distribution $p\left(y \mid \boldsymbol{x}\right)$ to $q\left(y \mid \boldsymbol{x}\right)$ is denoted as

$$D_{\mathrm{KL}}^{y|\boldsymbol{x}}(q \,\|\, p) \doteq \sum_{y=1}^{C} q\left(y \mid \boldsymbol{x}\right) \log(q\left(y \mid \boldsymbol{x}\right) / p\left(y \mid \boldsymbol{x}\right)).$$

Its expectation over $\boldsymbol{x} \sim p_{\boldsymbol{x}}$ is denoted $\mathbb{E}D_{\mathrm{KL}}^{y|\boldsymbol{x}}(q \,\|\, p)$ and is the main quantity of our study. The subscript r marks every variant attached to $\mathcal{D}_r$.

3.1 Formalism of the problem

For any first-order method mentioned in Sect. 2, the NN update $\boldsymbol{\ell}_{\boldsymbol{\theta}_u}(\boldsymbol{x}) - \boldsymbol{\ell}_{\boldsymbol{\theta}}(\boldsymbol{x})$ cannot be quantified in terms of unlearning efficacy: it is a function of the optimization trajectory and carries no information about the ideal classifier $p_r\left(y \mid \boldsymbol{x}\right)$. Our key idea is replacing it directly by the proxy shift $\log \hat{p}_r(\cdot \mid \boldsymbol{x}) - \log \hat{p}\left(\cdot \mid \boldsymbol{x}\right)$, the log-ratio of the two proxy posteriors, for which the shift is fully interpretable. Full details in App. C.1.

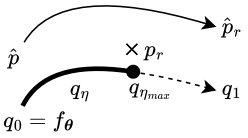


Figure 1: The unlearning signal is computed from the proxies $(\hat{p}, \hat{p}_r)$ and scaled by η to produce the unlearned family.

Unlearning with proxies. We look for a distribution $q\left(\cdot \mid \boldsymbol{x}\right)$, which is an update similar to the initial model $f_{\boldsymbol{\theta}}(\boldsymbol{x})$, closer to the ideal retain distribution p_r than the initial distribution p. This objective is framed in terms of KL divergences in the following optimization problem:

$$\min_q \; D_{\mathrm{KL}}^{y|\boldsymbol{x}}(q \,\|\, f_{\boldsymbol{\theta}}) \quad \text{s.t.} \quad D_{\mathrm{KL}}^{y|\boldsymbol{x}}(q \,\|\, p_r) \leq D_{\mathrm{KL}}^{y|\boldsymbol{x}}(q \,\|\, p)\,, \quad \sum_y q\left(y \mid \boldsymbol{x}\right) = 1. \tag{1}$$

An interpretation is the following. Among all distributions q that fall on the p_r side of the decision boundary (discriminating whether samples from q are more likely to be sampled from p or p_r), we seek the one closest to the initial model in relative entropy *i.e.* the most p-like distribution that the Neyman-Pearson likelihood test would still attribute in expectation to p_r instead of p.

Yet, p and p_r are unknown and cannot be used as such in the above problem (1). We thus introduce a pair of proxies $(\hat{p}, \hat{p}_r)$ for these two distributions. The formulation can be rewritten as:

$$\min_q \; D_{\mathrm{KL}}^{y|\boldsymbol{x}}(q \,\|\, f_{\boldsymbol{\theta}}) \quad \text{s.t.} \quad D_{\mathrm{KL}}^{y|\boldsymbol{x}}(q \,\|\, \hat{p}_r) \leq D_{\mathrm{KL}}^{y|\boldsymbol{x}}(q \,\|\, \hat{p})\,, \quad \sum_y q\left(y \mid \boldsymbol{x}\right) = 1. \tag{2}$$

The interpretation is the same as before, but now the decision boundary is defined by the proxies $(\hat{p}, \hat{p}_r)$ instead of the true distributions (p, p_r). One family that follows from it is, for a constant $\eta \geq 0$ and some normalization constant $\zeta(\boldsymbol{x}; \eta)$:

$$\log q_\eta(\cdot \mid \boldsymbol{x}) \doteq \boldsymbol{\ell}_{\boldsymbol{\theta}}(\boldsymbol{x}) + \eta \, (\log \hat{p}_r(\cdot \mid \boldsymbol{x}) - \log \hat{p}\left(\cdot \mid \boldsymbol{x}\right)) - \zeta(\boldsymbol{x}; \eta)\mathbf{1}. \tag{3}$$

Optimal scaling. We find the optimal value $\eta_{\max}$ as the upper bound of a theoretically safe regime (detailed in App. C.2). It is the largest scale at which the unlearning signal can be amplified while ensuring the theoretical bounds given by Prop. 1 and 2. In practice, no gradient nor back-propagation are required: $\eta_{\max}$ is computed as the next zero of a convex scalar function by 1-d bisection.

Optional distillation. The final proposed unlearned model is then this precise $q_{\eta_{\max}}$. The update $\eta_{\max}(\log \hat{p}_r - \log \hat{p})$ may be implemented as a logit processor on top of the initial classifier $\boldsymbol{\ell}_{\boldsymbol{\theta}}$ taken as a black-box provided it outputs a logit vector. An alternative is to distill the unlearning signal in the NN parameters $\boldsymbol{\theta}$ with the standard teacher-student approach (Hinton et al., 2015):

$$\boldsymbol{\theta}_u = \boldsymbol{\theta} + \arg\min_{\delta\boldsymbol{\theta}} \mathbb{E}D_{\mathrm{KL}}^{y|\boldsymbol{x}}(q_{\eta_{\max}} \,\|\, f_{\boldsymbol{\theta}+\delta\boldsymbol{\theta}})\,. \tag{4}$$

3.2 THEORETICAL GUARANTEES

This section gives a sketch of the reasoning, which is fully detailed in App. C and D.

The main quantity of the study, $\mathbb{E}D_{\mathrm{KL}}^{y|\boldsymbol{x}}(p_r \,\|\, q_\eta)$, gauges how close the probits q_η of the unlearned model are from the ideal distribution given by the conditional true distribution $p_r\left(y \mid \boldsymbol{x}\right)$. It can be exactly written as (see App. D.1):

$$\underbrace{\mathbb{E}D_{\mathrm{KL}}^{y|\boldsymbol{x}}(p_r \,\|\, f_{\boldsymbol{\theta}})}_{\text{initial divergence}} + \underbrace{\mathrm{Z}(\eta)}_{\leq 0, \forall \eta \in [0, \eta_{\max}]} + \eta \left(\underbrace{\mathbb{E}D_{\mathrm{KL}}^{y|\boldsymbol{x}}(p_r \,\|\, \hat{p}_r) - \mathbb{E}D_{\mathrm{KL}}^{y|\boldsymbol{x}}(p_r \,\|\, \hat{p})}_{\leq 0 \text{ if } \hat{p}_r \text{ models better } p_r \text{ than } \hat{p} \text{ does}} \right). \tag{5}$$

The derivation involves a specific normalizing term $\mathrm{Z}(\eta)$, which is a convex function of η with $\mathrm{Z}(0) = 0$. The value $\eta_{\max}$ is defined as the next zero of this function so that $\mathrm{Z}(\eta)$ is negative in the safety range $[0, \eta_{\max}]$ by design (see App. C.2). Finding $\eta_{\max}$ is a simple 1-d line search over the empirical evaluation of the convex function $\eta \mapsto \mathrm{Z}(\eta)$ (see Alg. 2 in App. C.2). The third term is negative if $\hat{p}_r$ is a better proxy for p_r than $\hat{p}$ is.

Proposition 1 (Provable decrease of KL for the unlearned family)**.** *For any $\eta \in [0, \eta_{\max}]$, if we assume that proxies are well designed i.e. $\mathbb{E}D_{\mathrm{KL}}^{y|\boldsymbol{x}}(p_r \,\|\, \hat{p}_r) \leq \mathbb{E}D_{\mathrm{KL}}^{y|\boldsymbol{x}}(p_r \,\|\, \hat{p})$ then*

$$\underbrace{\mathbb{E}D_{\mathrm{KL}}^{y|\boldsymbol{x}}(p_r \,\|\, q_\eta)}_{\textit{unlearned}} \leq \underbrace{\mathbb{E}D_{\mathrm{KL}}^{y|\boldsymbol{x}}(p_r \,\|\, f_{\boldsymbol{\theta}})}_{\textit{initial divergence}}. \tag{6}$$

This proposition is a soundness statement outlining a sufficient condition for a decrease. It asks $\hat{p}_r$ to track p_r better than what $\hat{p}$ does in expectation. In contrast, Prop. 2 does not require this assumption, but it involves the training error (see App D.2).

Proposition 2 (Tight bound on the divergence under small modeling shift)**.** *For any $\eta \in [0, \eta_{\max}]$, the main quantity $\mathbb{E}D_{\mathrm{KL}}^{y|\boldsymbol{x}}(p_r \,\|\, q_\eta)$ is tightly bounded by:*

$$(1-\eta) \underbrace{\mathbb{E}D_{\mathrm{KL}}^{y|\boldsymbol{x}}(p_r \,\|\, f_{\boldsymbol{\theta}})}_{\textit{initial divergence}} + \eta \left(\underbrace{\mathbb{E}D_{\mathrm{KL}}^{y|\boldsymbol{x}}(p_r \,\|\, \hat{p}_r) - \mathbb{E}D_{\mathrm{KL}}^{y|\boldsymbol{x}}(p \,\|\, \hat{p})}_{y|\boldsymbol{x} \textit{ modeling shift}} \right) + \eta \left(\underbrace{\mathbb{E}D_{\mathrm{KL}}^{y|\boldsymbol{x}}(p \,\|\, f_{\boldsymbol{\theta}})}_{\textit{training error}} + \mathcal{O}(\pi_f) \right), \tag{7}$$

where every $\mathbb{E}$ is taken under $\boldsymbol{x} \sim p_{\boldsymbol{x}}$ and $\mathcal{O}(\pi_f)$ is the explicit residual (31)*.*

The first term of the bound is close to zero either when the posteriors of p_r and p are similar (nothing to unlearn), or when η is close to 1. The second term of the bound can be small even if the proxies $(\hat{p}_r, \hat{p})$ may not accurately model their respective targets (p_r, p). A rule of thumb is to model both p and p_r similarly so their KL divergences compensate. Modeling only the retrain set as in Georgiev et al. (2025) is sub-optimal. The third term is the initial model's training error, which the unlearner cannot control. Inequality (7) is our tightest bound. It becomes an equality for $\eta = \eta_{\max}$ (see App. D.2). Section 5 sets $\eta = \eta_{\max}$ in every experiment to maximally reduce this bound. Of note, $\eta_{\max}$ might be null when the proxies are not sufficiently informative or when the difference between (p_r, p) is so small that there is nothing to unlearn, and the network remains untouched.

4 CONSTRUCTION OF UNLEARNING-AWARE PROXIES

This section proposes modeling the data distribution with a proxy $\hat{p}\left(\boldsymbol{x} \mid y\right)$, *i.e.* a parametric mathematical model of the true class-conditional distribution. This proxy induces the following posterior, whose logarithm plays the role of a 'logit' vector of $\mathbb{R}^C$:

$$\forall y \in [\![C]\!], \forall \boldsymbol{x} \in \mathbb{X}, \quad \hat{p}\left(y \mid \boldsymbol{x}\right) \doteq \frac{\hat{p}\left(\boldsymbol{x} \mid y\right) p(y)}{\sum_{c=1}^{C} \hat{p}\left(\boldsymbol{x} \mid c\right) p(c)}. \tag{8}$$

Appendix D.3 pursues the theoretical derivation for the new class-conditional data models $\boldsymbol{x} \mid y$.

Algorithm 1 MODELING with $2C$ lift. Pair $(\hat{p}, \hat{p}_r)$ obtained from a single fit on labels (y, s).

Require: forget set $\mathcal{D}_f$, retain set $\mathcal{D}_r$, feature map ϕ, any class-conditional family

$\boldsymbol{z} \leftarrow \phi(\boldsymbol{x})$ for every $\boldsymbol{x} \in \mathcal{D}$ ▷ proxies may live in feature space

$s(\boldsymbol{x}) \leftarrow r$ if $\boldsymbol{x} \in \mathcal{D}_r$, else f if $\boldsymbol{x} \in \mathcal{D}_f$ ▷ **2C lift**: labels are now $(y, s) \in [\![C]\!] \times \{r, f\}$

for all $(y, s) \in [\![C]\!] \times \{r, f\}$ **do**

  $\pi_s(y) \leftarrow {}^{|\mathcal{D}_s(y)|}/_{|\mathcal{D}(y)|}$ ▷ subclass prior

  data modeling $\hat{p}\,(\boldsymbol{z} \mid y, s)$

end for

$\hat{p}\,(y, s \mid \boldsymbol{z}) \leftarrow$ Bayes inversion (8) ▷ posterior over the $2C$ labels

$\hat{p}\,(y \mid \boldsymbol{z}) \leftarrow \hat{p}\,(y, r \mid \boldsymbol{z}) + \hat{p}\,(y, f \mid \boldsymbol{z})$ ▷ sum back to dimension C

$\hat{p}_r(y \mid \boldsymbol{z}) \leftarrow {}^{\hat{p}(y,r|\boldsymbol{z})}/_{\sum_{c=1}^{C} \hat{p}(y,r|\boldsymbol{z})}$ ▷ condition, do not refit

return $(\hat{p}, \hat{p}_r)$

Proposition 3 (Bound in terms of class-conditional data modeling errors)**.** *Regarding Prop. 2, the $y \mid \boldsymbol{x}$ modeling shift term, $\mathbb{E}D_{\mathrm{KL}}^{y|\boldsymbol{x}}(p_r \,\|\, \hat{p}_r) - \mathbb{E}D_{\mathrm{KL}}^{y|\boldsymbol{x}}(p \,\|\, \hat{p})$, can be rewritten as*

$$\underbrace{\mathbb{E}_r D_{\mathrm{KL}}^{\boldsymbol{x}|y}(p_r \,\|\, \hat{p}_r) - \mathbb{E}D_{\mathrm{KL}}^{\boldsymbol{x}|y}(p \,\|\, \hat{p})}_{\textit{data conditional modeling shift}} + \underbrace{D_{\mathrm{KL}}^{\boldsymbol{x}}(p \,\|\, \hat{p}) - D_{\mathrm{KL}}^{\boldsymbol{x}}(p_r \,\|\, \hat{p}_r)}_{\textit{raw data modeling shift}} + \mathcal{O}(\pi_f). \tag{9}$$

This highlights the advantage of using similar proxy data models for both sides of the unlearning signal, as already outlined in Prop 2.

4.1 THE PATH TO THE 2C CONSTRUCTION

The *naïve* application of this approach is to fit one class-conditional model per class, $\hat{p}$ on $\mathcal{D}$ and $\hat{p}_r$ on $\mathcal{D}_r$. A refinement assumes the following *mixture*: for each class y, $\hat{p}\,(\boldsymbol{x} \mid y) = (1 - \pi_f(y))\hat{p}_r(\boldsymbol{x} \mid y) + \pi_f(y)\hat{p}_f(\boldsymbol{x} \mid y)$, where $\pi_f(y) = {}^{|\mathcal{D}_f(y)|}/_{|\mathcal{D}(y)|}$ is the true intra-class proportion of the forget set.

The 2C lift. This refinement artificially considers the variable $(y, s) \in [\![C]\!] \times \{r, f\}$ as a label itself. This results in a mixture of $2C$ distributions. As a result, the proxy on the initial data intrinsically estimates $\hat{p}\,(s \mid \boldsymbol{x})$. This helps unlearning, especially when the forget data are not independent. To our knowledge, no other work proposes this lifting. This subtle shift results in a final mixture on the posterior rather than on $\boldsymbol{x} \mid y$. Alg. 1 details the mapping back to dimension C to create the final proxies.

This paradigm applies to any data model. Alg. 1 only requires a family that can be fitted per (y, s) and inverted by the Bayes rule (8). The Gaussian models of Sect. 4.2 and the empirical measure of Sect. 4.3 are two examples among mixtures, kernel densities, or any normalized density model. Another trick is to model the distribution of the features $\phi(\boldsymbol{x})$ rather than the raw input data $\boldsymbol{x}$. The feature map ϕ can be a frozen backbone or a projection onto a small subspace.

Section 5 reports results with parametric models because they make the procedure efficient and interpretable with no gradient nor hyper-parameter. Yet, this approach also applies to a learned $2C$ head on top of the feature map ϕ, reading $\hat{p}\,(y, s \mid \phi(\boldsymbol{x}))$ off its softmax.

4.2 APPLICATION TO GAUSSIAN PROXIES

Table 1 summarizes the three constructions with Gaussian models of $\boldsymbol{x} \mid y$ and links it to known discriminators `LDA` and `QDA` (see App. E). All of them need the same $2C$ estimations of expectation vectors, but they differ on the number of covariance matrix estimates. `LDA-2C` applies a unique `LDA` on $(y, s) \in [\![C]\!] \times \{r, f\}$. `QDA` remains unaffected by double labels.

4.3 APPLICATION TO THE EMPIRICAL MEASURE LEADS TO AN η-FREE BASELINE

Replacing the Gaussian distribution by the empirical measure ignores the features. The scale η disappears because $\log \hat{p}_r - \log \hat{p}$ takes values in $\{0, -\infty\}$, so any $\eta > 0$ gives the same target.

Table 1: Proxies based on the Gaussian distribution. Columns count the number of expectation vectors and covariance matrices to estimate the pair $(\hat{p}, \hat{p}_r)$.

	initial proxy $\hat{p}$	retain proxy $\hat{p}_r$	#$\boldsymbol{\mu}$	#$\boldsymbol{\Sigma}$
naive : the forget set is never looked at				
`LDA`	$\mathcal{N}(\boldsymbol{\mu}_y; \boldsymbol{\Sigma})$	$\mathcal{N}(\boldsymbol{\mu}_{r,y}; \boldsymbol{\Sigma}_r)$	$2C$	2
`QDA`	$\mathcal{N}(\boldsymbol{\mu}_y; \boldsymbol{\Sigma})$	$\mathcal{N}(\boldsymbol{\mu}_{r,y}; \boldsymbol{\Sigma}_{r,y})$	$2C$	$2C$
mixture : $\hat{p}$ is a mixture of a retain and a forget fit				
`LDA-Mix`	with $\hat{p}_f : \mathcal{N}(\boldsymbol{\mu}_{f,y}; \boldsymbol{\Sigma}_f)$	$\mathcal{N}(\boldsymbol{\mu}_{r,y}; \boldsymbol{\Sigma}_r)$	$2C$	2
`QDA-Mix`	with $\hat{p}_f : \mathcal{N}(\boldsymbol{\mu}_{f,y}; \boldsymbol{\Sigma}_{f,y})$	$\mathcal{N}(\boldsymbol{\mu}_{r,y}; \boldsymbol{\Sigma}_{r,y})$	$2C$	$2C$
$2C$ *lift : one fit per label (y, s), mixture at the posterior level*				
`LDA-2C`	$\mathcal{N}(\boldsymbol{\mu}_{s,y}; \boldsymbol{\Sigma})$	*conditioning*	$2C$	1

The empirical measure. `DIR` is the naive approach applied to the empirical measure, *i.e.* a sum of Diracs:

$$\hat{p} : (\boldsymbol{x}, y) \sim \mathcal{D}, \quad \hat{p}_r : (\boldsymbol{x}, y) \sim \mathcal{D}_r. \tag{10}$$

Appendix F shows that this creates an unlearned model s.t. $q(\boldsymbol{x}) = f_{\boldsymbol{\theta}}(\boldsymbol{x})$ if $\boldsymbol{x} \notin \mathcal{D}_f$, otherwise

$$q(\boldsymbol{x})_c = \begin{cases} 0 & c = y(\boldsymbol{x}), \\ f_{\boldsymbol{\theta}}(\boldsymbol{x})_c \,/\, (1 - f_{\boldsymbol{\theta}}(\boldsymbol{x})_{y(\boldsymbol{x})}) & c \neq y(\boldsymbol{x}). \end{cases} \tag{11}$$

Its main use is to be a legitimate teacher for the upcoming distillation in the case of a class-wise forgetting scenario due to the fact that it exactly annihilates the probit of $y(\boldsymbol{x})$ on the forget set. However, this is not suitable for a subclass forgetting scenario.

$2C$ lifting for Diracs. `DIR-2C` is the $2C$ lift of Sect. 4.1 applied to that same empirical measure:

$$\hat{p} : (\boldsymbol{x}, y, s) \sim \mathcal{D} \times \{r, f\}, \quad \hat{p}_r : y \mid \boldsymbol{x} \sim \hat{p}\,(y \mid \boldsymbol{x}, r)\,. \tag{12}$$

Its state posterior being feature agnostic, $\hat{p}\,(s \mid \boldsymbol{x}, y) \simeq \pi_s(y)$ (see App. F), the pair reduces to empirical counts. The same derivation leaves $q(\boldsymbol{x}) = f_{\boldsymbol{\theta}}(\boldsymbol{x})$ outside $\mathcal{D}_f$ and, on the forget set, gives:

$$q(\boldsymbol{x})_c = \frac{|\mathcal{D}_r(y(\boldsymbol{x}))|}{|\mathcal{D}(y(\boldsymbol{x}))|} f_{\boldsymbol{\theta}}(\boldsymbol{x})_c + \frac{|\mathcal{D}_f(y(\boldsymbol{x}))|}{|\mathcal{D}(y(\boldsymbol{x}))|} \begin{cases} 0 & c = y(\boldsymbol{x}), \\ f_{\boldsymbol{\theta}}(\boldsymbol{x})_c \,/\, (1 - f_{\boldsymbol{\theta}}(\boldsymbol{x})_{y(\boldsymbol{x})}) & c \neq y(\boldsymbol{x}), \end{cases} \tag{13}$$

where $|\mathcal{D}(y(\boldsymbol{x}))|$ is the number of training samples in $\mathcal{D}$ pertaining to the same class $y(\boldsymbol{x})$ than sample $\boldsymbol{x}$. Compared to (11), the same probit is now partially cancelled, in proportion of the forget set. This adapts `DIR` in the specific class-wise unlearning scenario. This is specifically what an agnostic unlearner would expect for behavior. Of note, `DIR-2C` is the cheapest proxy of the library, and the only one oblivious to the retain dataset. It only needs the count numbers $\{|\mathcal{D}_r(y)|\}_y$ and $\{|\mathcal{D}(y)|\}_y$.

5 Experiments

5.1 Experimental protocol

Datasets and architectures. The explored datasets are CIFAR-10 and CIFAR-100 (Krizhevsky, 2009). We consider (3+1) architectures. The first three use **DINOv2-Small** (Oquab et al., 2024) as a frozen backbone foundation model, completed with the feed forward heads: **linear** ($384 \to C$), **mlp1** ($384 \to 256 \to C$), or **mlp2** ($384 \to 256 \to 256 \to C$). The fourth is **ResNet-18**, where we apply our method to internal projected features (see App. H).

Unlearning methods. The results contain `LDA-2C`, the distilled `DIR` and its variant `DIR-2C` , `QDA` and `QDA-Mix` with diagonal covariances, `LDA` and `LDA-Mix` (see App. E). Our method is evaluated against iterative baselines trained for at most 20 epochs: **FT** (retain fine-tuning), **GA** (gradient ascent), **GA+FT** (one ascent followed by fine-tuning), **RL+FT** (Graves et al., 2021) (random labelling the forget set), **SCRUB** (Kurmanji et al., 2023) (min-max teacher-student), **SalUn** (Fan et al., 2024) (gradient ascent masking and RL+FT).

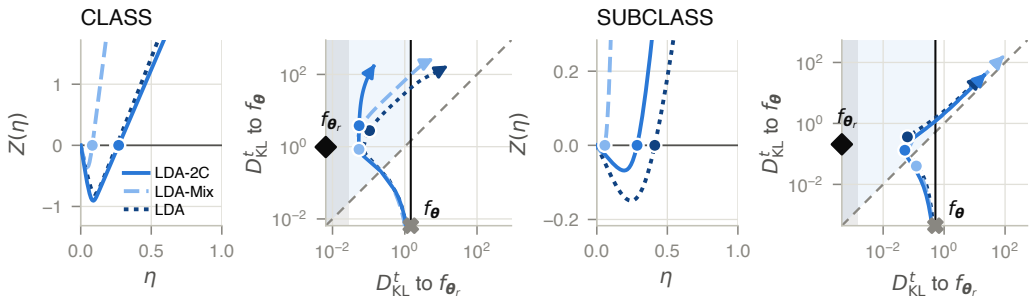


Figure 2: CIFAR-10, DINOv2 + MLP-1; *Class* and *Subclass* (airplane) scenarios. (left) $Z(\eta)$ for $\eta \in [0, 1]$; the dot is $\eta_{\max}$. (right) path of the target q_η in the plane $(D^{\mathrm{t}}_{\mathrm{KL}}(f_{\boldsymbol{\theta}_r}||q_\eta), D^{\mathrm{t}}_{\mathrm{KL}}(f_{\boldsymbol{\theta}}||q_\eta))$.

Unlearning scenarios. *Class* where $\mathcal{D}_f$ is an entire class. *Subclass* for CIFAR-10 consists in training a binary classifier (vehicle vs. animal) and forgetting one subclass per class at a time. For CIFAR-100 it is $C = 20$ classes, each composed of five subclasses. This scenario is difficult since the forget set is structured within the class. This causes gradient ascent baselines to fail Mavrothalassitis et al. (2025).

Metrics. $D^{\mathrm{t}}_{\mathrm{KL}}$ and $\mathrm{Acc}^{\mathrm{t}}$ (resp. $D^{\mathrm{f}}_{\mathrm{KL}}$ and $\mathrm{Acc}^{\mathrm{f}}$) measure $D^{y|\boldsymbol{x}}_{\mathrm{KL}}(f_{\boldsymbol{\theta}_r} \,\|\, q)$ and the accuracy empirically over the test set (resp. forget set). We also define the *noise floor* as the average $D^{y|\boldsymbol{x}}_{\mathrm{KL}}\left(f_{\boldsymbol{\theta}_r} \,\middle\|\, f_{\boldsymbol{\theta}_r}{}'\right)$ between two independently retrained models. This helps gauge how close an unlearned model is to models retrained from scratch. Per-sample membership inference audit with relative test, inspired by the work of Ribero et al. (2026)), as a percentage of data in the forget set whose outputs look more from retrain than from initial (see App. I.2). RTE is the runtime normalized by the duration of training $f_{\boldsymbol{\theta}_r}$ until it obtains 99% training accuracy.

5.2 Experimental validation

About $\eta_{\max}$. Figure 2 (left) empirically verifies two statements about the Z function: convexity and uniqueness of the zero $\eta_{\max}$ in $[0, 1)$. Alg. 2 has no difficulty in finding this zero, although its value depends a lot on the proxies and on the unlearning scenario. Figure 2(right) shows the parametric plot $D_{\mathrm{KL}}(f_{\boldsymbol{\theta}} \,\|\, q_\eta)$ *vs.* $D_{\mathrm{KL}}(f_{\boldsymbol{\theta}_r} \,\|\, q_\eta)$ as η goes from 0 to ∞. Model q_η starts from $f_{\boldsymbol{\theta}}$, gets closer to $f_{\boldsymbol{\theta}_r}$, and then goes away from both models. Along this trajectory, $\eta_{\max}$ almost always indicates the sweet spot closer to $f_{\boldsymbol{\theta}_r}$. The grey area represents the noise floor. The blue area depicts desirable unlearn models which are closer to $f_{\boldsymbol{\theta}_r}$ than to the initial $f_{\boldsymbol{\theta}}$ and closer to $f_{\boldsymbol{\theta}_r}$ than $f_{\boldsymbol{\theta}}$ is.

Getting closer to the ideal. Figure 3 compares the architectures DINOv2 and ResNet-18 to outline the dependence on the features. Our $\eta_{\max}$ construction almost always gets closer to the ideal model as promised by Prop. 1. This is striking for DINOv2 but less tangible for ResNet-18. However, our targets are mostly on the edge of the noise floor. Competitors cannot achieve this goal with certainty.

Comparison between proxies. Fig. 5 illustrates the benefits of the 2C construction: `DIR` destroys the entire class, and its distillation does not verify Prop. 1, while `DIR-2C` does. Figure 4 shows the drawback of `LDA-Mix`, the only proxy that estimates a full covariance matrix for $\mathcal{D}_f$ alone. On the contrary, `LDA` and `LDA-2C` use a single pooled covariance matrix estimated across all classes.

5.3 Benchmark

Behavioral advantage. Our proxies are competitive in the *Class* scenario, matching the ideal model's accuracy exactly over the forget set and within the accuracy noise floor on the test set (see Fig. 4). As for *Subclass*, they reach the lowest KL well below the baselines (see Fig. 5) at a much smaller RTE (see Table. 2). On DINOv2, our approach is the best affordable unlearning procedure.

Distillation and stopping criterion. Figure 5 shows that the distillation, which is an optional final step, slightly improves the behavior. We explain this by the fact that the distilled $f_{\boldsymbol{\theta}_u}$ shares the same architecture as $f_{\boldsymbol{\theta}}$. Distillation requires no rewind or stopping criterion; it converges quickly and

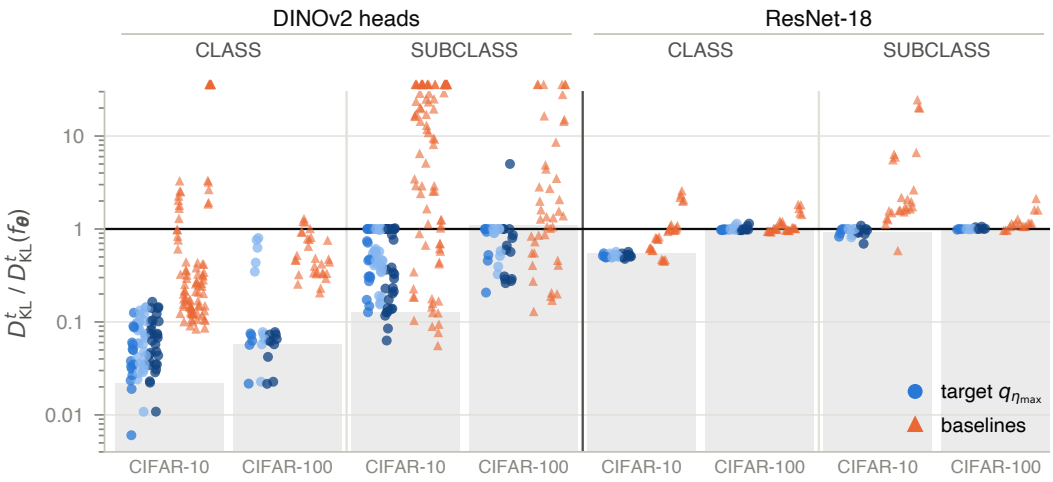


Figure 3: Empirical validation of Prop. 1. $D^{\mathrm{t}}_{\mathrm{KL}}/D^{\mathrm{t}}_{\mathrm{KL}}(f_{\boldsymbol{\theta}})$ for 4 architectures, mean over 5 seeds. (blue): `LDA-2C`, `LDA-Mix`, `QDA-Mix`, `LDA`, `QDA`; (orange): the six baselines at their last epoch.

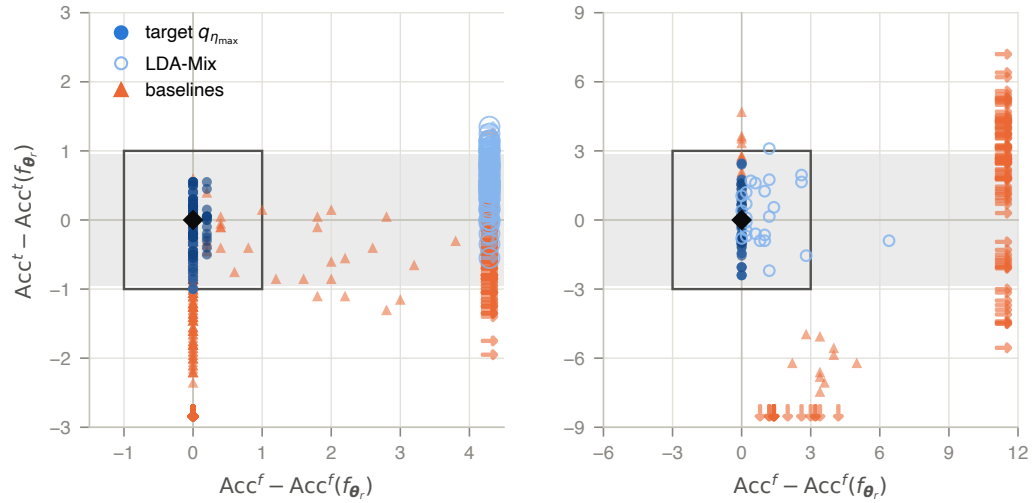


Figure 4: $\mathrm{Acc_f} - \mathrm{Acc_f}(f_{\boldsymbol{\theta}_r})$ *vs.* $\mathrm{Acc_t} - \mathrm{Acc_t}(f_{\boldsymbol{\theta}_r})$, *Class* scenario. square: tolerance ± 1 pt (DINOv2 heads-left), ± 3 pt (ResNet-18-right); grey band: accuracy noise floor over the retrain seeds.

remains stable. In contrast, ascent-based methods GA and GA+FT deviate from expected behavior. The others plateau or converge very slowly.

6 Conclusion: Advantages and Limitations

Our framework introduces a closed-form unlearning operator emerging from an explicit constrained optimization problem (2). It derives the unlearning signal from any proxies and adapts its scaling in a safe regime. This approach can be seen as a logit processor on top of the initial black-box model. Updating the weights by distillation is optional, fast, and stable.

Prop. 1 and 2 are general, while Prop. 3 deals with data modeling. They provide *behavioral* guarantees that bound the divergence to the ideal distribution, not a retrained model. The experiments show that this indeed translates to the retrained model.

The paper instantiates this approach with a family of proxies without claiming optimality. More useful are the general 2C construction and the dimension-reduction projection, which apply to any

Table 2: Every method gets the same budget, 10% of the retraining, read at the last epoch (Ep.) it can afford. CIFAR-10 / DINOv2+`mlp1`, *Subclass* (airplane), mean±std over 5 seeds. The targets $q_{\eta_{\max}}$ take no gradient step so RTE is the proxy fit. **Bold**: closest to $f_{\boldsymbol{\theta}_r}$ in the columns.

	$\mathrm{Acc}^{\mathrm{f}}$ (%)	$\mathrm{Acc}^{\mathrm{t}}$ (%)	$D_{\mathrm{KL}}^{\mathrm{f}}$ (nats)	$D_{\mathrm{KL}}^{\mathrm{t}}$ (nats)	Audit (%)	RTE (%)	Ep.
$f_{\boldsymbol{\theta}}$	100.0	99.8	5.0	0.49	0.0	–	–
$f_{\boldsymbol{\theta}_r}$	59.1	95.8	0.010	$1.1 \cdot 10^{-3}$	100.0	100	–
`LDA-2C`	**61.9** ± 0.1	**96.3** ± 0.1	**0.48** ± 0.02	**0.062** ± 0.003	59.3 ± 1.1	4.7	0
`LDA-Mix`	92.3 ± 0.1	99.1 ± 0.1	1.7 ± 0.1	0.20 ± 0.01	14.2 ± 0.7	3.0	0
`LDA`	45.8 ± 0.2	95.0 ± 0.0	0.61 ± 0.04	**0.062** ± 0.004	**72.6** ± 0.5	3.0	0
`QDA-Mix`	69.9 ± 0.1	97.2 ± 0.1	0.63 ± 0.02	0.079 ± 0.003	49.3 ± 1.3	3.3	0
`QDA`	33.7 ± 0.4	93.9 ± 0.1	0.96 ± 0.05	0.099 ± 0.006	**72.8** ± 0.5	3.0	0
SCRUB	4.1 ± 3.6	90.2 ± 0.7	29 ± 20	2.6 ± 1.8	48.4 ± 9.6	7.7	3
GA+FT	6.8 ± 1.4	90.8 ± 0.2	25 ± 4	2.1 ± 0.4	44.9 ± 1.5	7.8	2
GA	0.0 ± 0.0	60.8 ± 0.0	$(2.4 \pm 0.1) \cdot 10^3$	745 ± 29	17.0 ± 0.6	7.7	20
SalUn	0.8 ± 0.2	90.3 ± 0.0	7.0 ± 0.6	0.60 ± 0.06	56.0 ± 1.2	8.0	2
RL+FT	1.2 ± 0.1	90.3 ± 0.0	9.5 ± 0.7	0.81 ± 0.06	53.6 ± 1.1	9.5	13
FT	95.7 ± 2.3	99.5 ± 0.2	2.7 ± 0.4	0.27 ± 0.05	6.3 ± 3.4	7.7	3

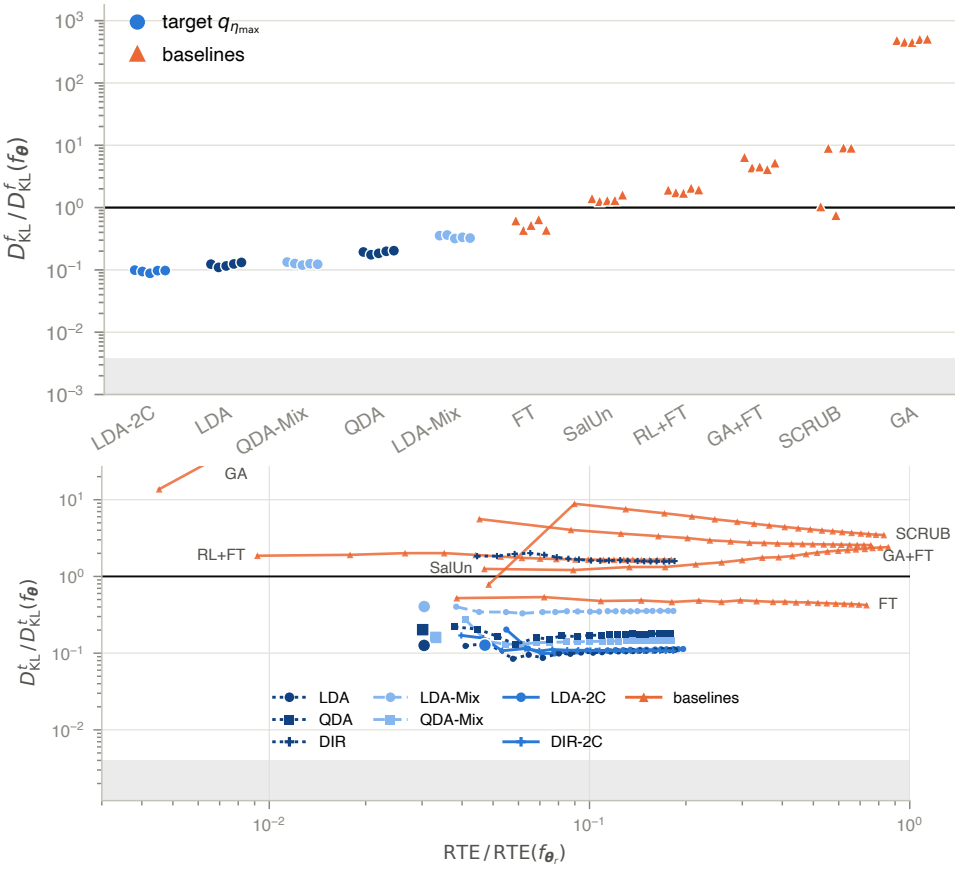


Figure 5: CIFAR-10, DINOv2 + MLP-1, *Subclass* (airplane). $D_{\mathrm{KL}}^{\mathrm{t}} / D_{\mathrm{KL}}^{\mathrm{t}}(f_{\boldsymbol{\theta}})$. (top) final result for 5 runs; (bottom) averaged results along the unlearning epochs *vs.* the cumulative RTE.

proxy. If the design of the proxies is wrong, it sets the scaling to zero, leaving the model unchanged. In contrast, the previous approach may destroy the model by unlearning too much.

7 REPRODUCIBILITY STATEMENT

For sake of reproducibility, we share the anonymous github link `https://anonymous.4open.science/r/unlearning-guarantees-7003`, which contains all the codes for generating the results presented in our article. This includes our implementation the SOTA baselines and smoke tests. See the file `README.md`

Remarks:

- Every source of randomness is seeded: weights and batch order (s), RANDOM splits (s), evaluation subsamples (seed 0), ResNet JL projections (seed 0). Each (cell, seed, method) run is independent of what else runs in the process.
- Same hardware and software $\rightarrow$ bit-identical results. Across hardware, float rounding (cuDNN / BLAS) differs, so numbers agree statistically, not bitwise. The paper's data were produced on CUDA GPUs.
- The proxy fits runs in torch on CUDA and in numpy on CPU/MPS: same math, different rounding.
- The paper's DINOv2 features were extracted on Apple MPS; re-extracting there gives the same files bit for bit, on CPU/CUDA to within $\approx 1e-4$.
- Linear heads reach the same model across hardware (logits within $\approx$ 1e-3); MLP heads and ResNets need not (logit differences of order 1 between CUDA and MPS training were observed on MLP heads). The signal set is defined by the reference models, so on a different machine it can change by a few percent of the forget set, which shifts every `*_signal` column of a cell together. The unrestricted columns, accuracies, KLs and $\eta_{\max}$ reproduce within seed noise.
- Runtimes are wall-clock; compare them only between files with the same `_machine` stamp (`epochs.json` $\rightarrow$ `rte.cross_node` lists mismatches).

When run on a V100/A100 cluster, the full batch of experiments costs 7.5 GPU-hours for DinoV2 architecture, 51 GPU-hours for Resnet architecture, producing a total of 8GB of results.

8 AI DISCLOSURE

We have **not** used generative AI tools for: *help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, assist with translation, clean and reformat dataset, support qualitative and thematic data analysis, interpret results*, or *generate synthetic data sets*.

Additionally, we used generative AI tools for *creating parts of the software code* (especially the scientific figures with Claude Code), and for *editing the research paper to improve readability* (Grammarly).

We have reviewed all AI-assisted work. LLM-generated code was verified and tested for correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

G. Alain and Y. Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

S.-I. Amari. A foundation of information geometry. *Electronics and Communications in Japan (Part I: Communications)*, 66(6):1–10, 1983. doi: https://doi.org/10.1002/ecja.4400660602. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/ecja.4400660602.

J. L. Bentley and A. C.-C. Yao. An almost optimal algorithm for unbounded searching. *Inf. Process. Lett.*, 5:82–87, 1976. URL https://api.semanticscholar.org/CorpusID:42725295.

J. L. Block, M. Mohri, A. Mokhtari, and S. Shakkottai. Temper-then-tilt: Principled unlearning for generative models through tempering and classifier guidance. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*. PMLR, 2026. URL https://arxiv.org/abs/2602.10217.

L. Bourtoule, V. Chandrasekaran, C. A. Choquette-Choo, H. Jia, A. Travers, B. Zhang, D. Lie, and N. Papernot. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (S&P)*, pages 141–159. IEEE, 2021. doi: 10.1109/SP40001.2021.00019.

Y. Cao and J. Yang. Towards making systems forget with machine unlearning. In *2015 IEEE Symposium on Security and Privacy*, pages 463–480. IEEE, 2015.

N. Carlini, S. Chien, M. Nasr, S. Song, A. Terzis, and F. Tramèr. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (S&P)*, pages 1897–1914. IEEE, 2022. doi: 10.1109/SP46214.2022.9833649.

M. Chen, W. Gao, G. Liu, K. Peng, and C. Wang. Boundary unlearning: Rapid forgetting of deep networks via shifting the decision boundary. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 06 2023. doi: 10.1109/CVPR52729.2023.00750.

E. Chien, H. Wang, Z. Chen, and P. Li. Langevin unlearning: a new perspective of noisy gradient descent for machine unlearning. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NeurIPS, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385.

V. S. Chundawat, A. K. Tarun, M. Mandal, and M. Kankanhalli. Can bad teaching induce forgetting? unlearning in deep networks using an incompetent teacher. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press, 2023. ISBN 978-1-57735-880-0. doi: 10.1609/aaai.v37i6.25879. URL https://doi.org/10.1609/aaai.v37i6.25879.

T. Cover and J. Thomas. *Elements of information theory*. Wiley-Interscience, 2006.

I. Csiszár. i-divergence geometry of probability distributions and minimization problems. *The Annals of Probability*, 3(1):146–158, 1975. doi: 10.1214/aop/1176996454.

S. Dasgupta and A. Gupta. An elementary proof of a theorem of Johnson and Lindenstrauss. *Random Structures & Algorithms*, 22(1):60–65, 2003.

E. Daxberger, A. Kristiadi, A. Immer, R. Eschenhagen, M. Bauer, and P. Hennig. Laplace redux – effortless Bayesian deep learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pages 20089–20103. Curran Associates, Inc., 2021.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–38, 1977.

V. Dine, T. Furon, and C. Faure. Improving unlearning with model updates probably aligned with gradients. In *Proceedings of the 18th ACM Workshop on Artificial Intelligence and Security*, AISec '25, page 158–169, New York, NY, USA, 2026. Association for Computing Machinery. ISBN 9798400718953. doi: 10.1145/3733799.3762975. URL https://doi.org/10.1145/3733799.3762975.

C. Dwork and A. Roth. *The Algorithmic Foundations of Differential Privacy*, volume 9 of *Foundations and Trends in Theoretical Computer Science*. Now Publishers, 2014.

R. Eldan and M. Russinovich. Who's Harry Potter? approximate unlearning in LLMs. *arXiv preprint arXiv:2310.02238*, 2023.

C. Fan, J. Liu, Y. Zhang, E. Wong, D. Wei, and S. Liu. SalUn: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. In *The 12th International Conference on Learning Representations (ICLR)*, 2024. Spotlight.

J. Foster, S. Schoepf, and A. Brintrup. Fast machine unlearning without retraining through selective synaptic dampening. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence (AAAI)*, pages 12043–12051. AAAI Press, 2024. doi: 10.1609/aaai.v38i11.29092.

J. H. Friedman. Regularized discriminant analysis. *Journal of the American Statistical Association*, 84(405):165–175, 1989.

R. Gandikota, J. Materzyńska, J. Fiotto-Kaufman, and D. Bau. Erasing concepts from diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2426–2436, 2023.

A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin. *Bayesian Data Analysis*. Chapman and Hall/CRC, Boca Raton, FL, 3rd edition, 2013.

K. G. Georgiev, R. Rinberg, S. Park, S. Garg, A. Ilyas, A. Madry, and S. Neel. Machine unlearning via simulated oracle matching. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu, editors, *International Conference on Learning Representations*, volume 2025, pages 49693–49731, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/7c799b09cc40973ceaa47da50131dc63-Paper-Conference.pdf.

Z. Ghahramani and G. E. Hinton. The EM algorithm for mixtures of factor analyzers. Technical Report CRG-TR-96-1, Department of Computer Science, University of Toronto, 1996.

A. Ginart, M. Y. Guan, G. Valiant, and J. Y. Zou. Making AI forget you: Data deletion in machine learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3513–3526, 2019.

S. Goel, A. Prabhu, A. Sanyal, S.-N. Lim, P. Torr, and P. Kumaraguru. Towards adversarial evaluations for inexact machine unlearning. *arXiv preprint arXiv:2201.06640*, 2022.

S. Goel, A. Prabhu, P. Torr, P. Kumaraguru, and A. Sanyal. Corrective machine unlearning. *arXiv preprint arXiv:2402.14015*, 2024.

A. Golatkar, A. Achille, and S. Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9304–9312, 2020a.

A. Golatkar, A. Achille, and S. Soatto. Forgetting outside the box: Scrubbing deep networks of information accessible from input–output observations. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 383–398. Springer, 2020b.

A. Golatkar, A. Achille, A. Ravichandran, M. Polito, and S. Soatto. Mixed-privacy forgetting in deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 792–801. IEEE, 2021.

L. Graves, V. Nagisetty, and V. Ganesh. Amnesiac machine learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

C. Guo, T. Goldstein, A. Hannun, and L. van der Maaten. Certified data removal from machine learning models. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020.

J. Hayes, I. Shumailov, E. Triantafillou, A. Khalifa, and N. Papernot. Inexact unlearning needs more careful evaluations to avoid a false sense of privacy. In *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pages 497–519, 04 2025. doi: 10.1109/SaTML64287.2025.00034.

G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network, 2015. URL https://arxiv.org/abs/1503.02531.

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

J. Ji, Y. Liu, Y. Zhang, G. Liu, R. R. Kompella, S. Liu, and S. Chang. Reversing the forget-retain objectives: An efficient LLM unlearning framework from logit difference. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37. Curran Associates, Inc., 2024. doi: 10.52202/079017-0400. URL https://arxiv.org/abs/2406.08607.

J. Jia, J. Liu, P. Ram, Y. Yao, G. Liu, Y. Liu, P. Sharma, and S. Liu. Model sparsity can simplify machine unlearning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. Spotlight.

W. B. Johnson and J. Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary Mathematics*, 26:189–206, 1984.

S. Kodge, G. Saha, and K. Roy. Deep unlearning: Fast and efficient gradient-free approach to class forgetting, 2024. URL https://arxiv.org/abs/2312.00761.

P. W. Koh and P. Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 1885–1894. PMLR, 2017.

A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

A. Kumar, A. Raghunathan, R. Jones, T. Ma, and P. Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *The 10th International Conference on Learning Representations (ICLR)*, 2022.

M. Kurmanji, P. Triantafillou, J. Hayes, and E. Triantafillou. Towards unbounded machine unlearning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36. Curran Associates, Inc., 2023.

A. Le, C. Peng, Y. Liu, and J. A. Noble. POUR: A provably optimal method for unlearning representations via neural collapse. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10273–10282, 2026. URL https://arxiv.org/abs/2511.19339.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

N. Li, A. Pan, A. Gopal, S. Yue, D. Berrios, A. Gatti, J. D. Li, A.-K. Dombrowski, S. Goel, L. Phan, et al. The WMDP benchmark: Measuring and reducing malicious use with unlearning. In *International Conference on Machine Learning (ICML)*, 2024.

S. Liu, Y. Yao, J. Jia, S. Casper, N. Baracaldo, P. Hase, Y. Yao, C. Y. Liu, X. Xu, H. Li, et al. Rethinking machine unlearning for large language models. *Nature Machine Intelligence*, 2025. doi: 10.1038/s42256-025-00985-0.

Z. Liu, Y. Jiang, J. Shen, M. Peng, K.-Y. Lam, X. Yuan, and X. Liu. A survey on federated unlearning: Challenges, methods, and future directions. *ACM Computing Surveys*, 2024. doi: 10.1145/3679014.

P. Maini, Z. Feng, A. Schwarzschild, Z. C. Lipton, and J. Z. Kolter. TOFU: A task of fictitious unlearning for LLMs. In *Conference on Language Modeling (COLM)*, 2024.

I. Mavrothalassitis, P. Puigdemont, N. I. Levi, and V. Cevher. Ascent fails to forget. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2025. URL https://openreview.net/forum?id=KBJSV1XApq.

G. J. McLachlan. *Discriminant Analysis and Statistical Pattern Recognition*. Wiley-Interscience, wiley classics library edition, 1992.

G. J. McLachlan and D. Peel. *Finite Mixture Models*. Wiley, New York, 2000.

R. Mehta, S. Pal, V. Singh, and S. N. Ravi. Deep unlearning via randomized conditionally independent hessians. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10422–10431, 2022.

K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, Cambridge, MA, 2012.

N. Naderloui, S. Yan, B. Wang, J. Fu, W. H. Wang, W. Liu, and Y. Hong. Rectifying privacy and efficacy measurements in machine unlearning: A new inference attack perspective. In *34th USENIX Security Symposium (USENIX Security 25)*. USENIX Association, 2025. URL https://arxiv.org/abs/2506.13009.

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

Q. P. Nguyen, B. K. H. Low, and P. Jaillet. Variational Bayesian unlearning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

T. T. Nguyen, T. T. Huynh, P. L. Nguyen, A. W.-C. Liew, H. Yin, and Q. V. H. Nguyen. A survey of machine unlearning. *arXiv preprint arXiv:2209.02299*, 2022.

M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, et al. DINOv2: Learning robust visual features without supervision. In *Transactions on Machine Learning Research*, 2024.

V. Papyan, X. Y. Han, and D. L. Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40):24652–24663, 2020. doi: 10.1073/pnas.2015509117.

A. Peste, D. Alistarh, and C. H. Lampert. SSSE: Efficiently erasing samples from trained machine learning models. In *NeurIPS 2021 Workshop on Privacy in Machine Learning (PrivateML)*, 2021.

M. Ribero, A. Schrab, and A. Gretton. Regularized f-divergence kernel tests. In *AISTATS*, 2026. URL https://arxiv.org/abs/2601.19755.

A. Sekhari, J. Acharya, G. Kamath, and A. T. Suresh. Remember what you want to forget: Algorithms for machine unlearning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pages 18075–18086. Curran Associates, Inc., 2021.

R. Shokri, M. Stronati, C. Song, and V. Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (S&P)*, pages 3–18. IEEE, 2017. doi: 10.1109/SP.2017.41.

V. M. Suriyakumar, A. Sekhari, and A. Wilson. UCD: Unlearning in LLMs via contrastive decoding, 2025. URL https://arxiv.org/abs/2506.12097.

A. K. Tarun, V. S. Chundawat, M. Mandal, and M. S. Kankanhalli. Fast yet effective machine unlearning. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, 2023. doi: 10.1109/TNNLS.2023.3266828.

A. Thudi, G. Deza, V. Chandrasekaran, and N. Papernot. Unrolling SGD: Understanding factors influencing machine unlearning. In *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*, pages 303–319. IEEE, 2022a.

A. Thudi, H. Jia, I. Shumailov, and N. Papernot. On the necessity of auditable algorithmic definitions for machine unlearning. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 4007–4022, 2022b.

E. Triantafillou, P. Kairouz, F. Pedregosa, J. Hayes, M. Kurmanji, K. Zhao, V. Dumoulin, J. J. Junior, I. Mitliagkas, J. Wan, L. S. Hosoya, S. Escalera, G. K. Dziugaite, P. Triantafillou, and I. Guyon. Are we making progress in unlearning? findings from the first neurips unlearning competition, 2024. URL `https://arxiv.org/abs/2406.09073`.

P. Voigt and A. Von dem Bussche. *The EU General Data Protection Regulation (GDPR): A Practical Guide*. Springer International Publishing, Cham, 1st edition, 2017.

A. Warnecke, L. Pirch, C. Wressnegger, and K. Rieck. Machine unlearning of features and labels. In *Proceedings of the 30th Network and Distributed System Security Symposium (NDSS)*, 2023.

H. Xu, T. Zhu, L. Zhang, W. Zhou, and P. S. Yu. Machine unlearning: A survey. *ACM Computing Surveys*, 56(1):1–36, 2023. doi: 10.1145/3603620.

S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, pages 268–282. IEEE, 2018. doi: 10.1109/CSF.2018.00027.

S. Zhou, T. Yu, Z. Zhang, H. Chang, X. Zhou, D. Wu, and H. Zhao. Efficient utility-preserving machine unlearning with implicit gradient surgery. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

A MACHINE UNLEARNING OVERVIEW

This appendix expands the related work presented in Sect. 2.

A.1 MACHINE UNLEARNING

The literature is now broad enough that several survey papers cover the field from complementary angles (Nguyen et al., 2022; Xu et al., 2023; Liu et al., 2024; 2025).

Motivation and regulatory context. The renewed interest in machine unlearning is driven by data-protection regulations such as the EU General Data Protection Regulation (GDPR) and its "right to be forgotten" (Voigt and Von dem Bussche, 2017), together with the California Consumer Privacy Act (CCPA) and similar frameworks worldwide. Beyond legal compliance, unlearning has been also identified as a tool for removing the influence of poisoned, biased, mislabelled, copyrighted or otherwise undesirable training samples (Cao and Yang, 2015; Bourtoule et al., 2021; Goel et al., 2024), and as a primitive for AI safety (e.g., suppressing knowledge in foundation models (Li et al., 2024)).

Exact unlearning. Machine unlearning was formalised by Cao and Yang (2015), who introduced the term and proposed summation-form algorithms that allow data removal without full retraining. The gold standard remains exact removal, in which the unlearned model is statistically indistinguishable from a model retrained from scratch on the retained set (Cao and Yang, 2015; Bourtoule et al., 2021; Ginart et al., 2019). Bourtoule et al. (2021) achieve exact unlearning in deep models via SISA training, which partitions the training data into shards so that only the affected shard needs to be retrained on deletion. Although SISA reduces retraining cost by isolating deletions, it requires storing intermediate checkpoints for every shard, and at scale this storage overhead and the accuracy loss caused by aggressive sharding make it impractical for large deep networks.

Gradient-based approximate unlearning. To avoid retraining, the dominant paradigm relies on gradient-based weight updates. Thudi et al. (2022a) provide an unrolled-SGD analysis of the verification error and show how a training-time penalty can ease subsequent unlearning. Knowledge-distillation variants jointly enforce forgetting and retention: SCRUB (Kurmanji et al., 2023) trains a student that minimises the KL divergence to the original model on the retained data while maximising the loss on the forget set. Chundawat et al. (2023) use an incompetent teacher whose soft labels guide the student to suppress class knowledge. SalUn (Fan et al., 2024) restricts updates to the weights identified as most responsible for the forgotten data via gradient-based saliency, reaching state-of-the-art results on image classification and generation. Jia et al. (2023) show that model sparsification prior to unlearning closes the gap between exact and approximate unlearning across multiple algorithms. Tarun et al. (2023) propose UNSIR, an impair-repair scheme in which an error-maximising noise first degrades class knowledge and a repair step then restores retain-set accuracy. Neel et al. (2021) provide gradient-descent-based deletion algorithms that handle adversarial sequences of deletion requests for convex models. Beyond their empirical strengths, these first-order methods share a structural limitation: the update follows an optimisation trajectory that admits no closed-form characterisation in terms of the data distribution or the resulting predictive shift.

Fisher information and synaptic dampening. Fisher-based methods identify which parameters encode the forgotten data and selectively modify them. Golatkar et al. (2020a) derive a Fisher Information Matrix (FIM) scrubbing procedure that perturbs weights in order to remove cohort information, and Golatkar et al. (2020b) extend this analysis via a Neural Tangent Kernel linearisation that better handles the null-space of the weights and bounds the information accessible from input-output observations. Foster et al. (2024) propose Selective Synaptic Dampening (SSD), a retrain-free two-step method that uses the FIMs of the training and forget sets to select and dampen the parameters most important to the forget set. SSD is computationally light and does not require retained data at unlearning time, but its forgetting guarantee is coupled to the quality of the diagonal FIM approximation, which does not expose the structure of the data distribution. Mehta et al. (2022) introduce L-CODEC, a randomised conditional-independence test that selects a subset of the model's parameters, making Newton-style updates tractable for very large vision and NLP models.

Influence functions and certified removal. Influence functions (Koh and Liang, 2017), borrowed from robust statistics, approximate the leave-one-out parameter shift via a single Newton step on the training loss. Guo et al. (2020) exploit this connection to derive certified data removal for ℓ_2-regularised linear models: a calibrated noise mechanism masks the residual Newton-step error, yielding an (ϵ, δ) differential-privacy certificate. Sekhari et al. (2021) extend the theoretical analysis by bounding generalisation after unlearning, and Izzo et al. (2021) propose a projection-based deletion algorithm whose cost is linear in the feature dimension and independent of the dataset size for linear and logistic models. Warnecke et al. (2023) extend influence-based unlearning to the deletion of features and labels, providing closed-form updates and certified guarantees for strongly convex losses. Beyond linear models, extending influence-function guarantees to non-convex deep networks remains challenging (Koh and Liang, 2017; Guo et al., 2020): the Hessian approximation is hard to control, and no closed-form bound on the predictive shift in terms of the data distribution is available. More recent work such as Chien et al. (2024) uses Langevin dynamics to obtain certified unlearning without convexity assumptions, but the resulting bounds still ignore the class-conditional structure of the data.

Weight-space and Bayesian methods. A complementary line of work reasons about model uncertainty in parameter space rather than via optimisation trajectories. Golatkar et al. (2021) extend FIM scrubbing to a mixed-privacy setting by targeting only non-core data, and Peste et al. (2021) scale FIM-based scrubbing to deep networks via rank-one Hessian approximations. The Laplace approximation (Daxberger et al., 2021) provides a principled Gaussian posterior over weights and has been adopted in Bayesian unlearning to characterise the uncertainty introduced by data deletion (Nguyen et al., 2020). All of these methods operate entirely in weight space: their error guarantees are coupled to Hessian-approximation quality and do not decompose in terms of the quality of the class-conditional data model.

Geometry-based and training-free unlearning. A recent direction exploits the geometric structure of trained representations to obtain closed-form, training-free corrections. Le et al. (2026) build on Neural Collapse theory (Papyan et al., 2020) to show that the orthogonal projection of a simplex Equiangular Tight Frame (ETF) onto the subspace of retained classes remains an ETF in the lower-dimensional space, which yields a provably optimal forgetting operator. Their method, POUR-P, delivers this correction in closed form and without gradient steps. In a similar spirit, Kodge et al. (2024) project model weights onto the subspace orthogonal to the principal directions of the forget concepts, rendering the model blind to that information without retraining. While these methods are fast and elegant, the optimality guarantees of projection-based approaches such as POUR rely on the terminal neural-collapse regime, and neither family provides a data-distribution bound on the residual unlearning error.

Theoretical guarantees and certification criteria. Guo et al. (2020); Sekhari et al. (2021) measure unlearning success as (ϵ, δ)-indistinguishability, mirroring differential privacy (Dwork and Roth, 2014). Total variation and KL divergence between the ideal retrained model and the empirically unlearned model have since been adopted as natural criteria (Golatkar et al., 2020a; Chien et al., 2024), but nearly all theoretical results are stated in terms of algorithmic properties (Hessian approximation quality, loss Lipschitz constants) rather than in terms of the data distribution. A separate line of work studies the auditability of unlearning definitions and the difficulty of certifying forgetting in arbitrary training pipelines (Thudi et al., 2022b), underlining the gap between algorithmic and distributional guarantees. To our knowledge, no prior work provides a closed-form decomposition of the expected KL divergence directly in terms of the quality of distributional shift of models fitted to the data (see Prop. 2).

Membership inference as an evaluation tool. Membership inference attacks (MIAs) were introduced by Shokri et al. (2017), formally connected to overfitting by Yeom et al. (2018), and sharpened by the likelihood-ratio attack LiRA of Carlini et al. (2022), which achieves the highest statistical power at low false-positive rates thanks to held-out shadow models. MIAs have become the standard black-box audit for unlearning efficacy (Triantafillou et al., 2024; Hayes et al., 2025): an effective unlearning should prevent an adversary from distinguishing forgotten examples from held-out non-members. Hayes et al. (2025) show that population-level MIAs systematically overestimate the privacy protection of approximate methods, and they propose U-LiRA, a per-example extension that serves as a stricter worst-case audit. Complementary evaluation protocols have also been developed:

Goel et al. (2022) introduce the Interclass Confusion test together with the EU-k and CF-k baselines, and the findings of the first NeurIPS unlearning competition (Triantafillou et al., 2024) provide a broad empirical assessment of the state of the art under a unified evaluation pipeline.

Unlearning beyond classification: generative models and LLMs. Although the present work focuses on classification, machine unlearning has recently been extended to generative models, where it is used to suppress copyrighted, sensitive or harmful content. For diffusion models, Gandikota et al. (2023) propose Erased Stable Diffusion (ESD), a fine-tuning method that removes a target concept from the model weights using only a textual description and negative guidance. SalUn (Fan et al., 2024) also covers concept erasure in diffusion models. For large language models, Eldan and Russinovich (2023) introduce the "Who's Harry Potter?" protocol for unlearning copyrighted content. Maini et al. (2024) propose the TOFU benchmark of fictitious authors for controlled evaluation, and Li et al. (2024) introduce the WMDP benchmark for hazardous-knowledge unlearning together with the RMU representation misdirection method. These efforts highlight that distributional unlearning guarantees are even more elusive in the generative regime, since the relevant "data distribution" is itself implicit in the model.

B GLOSSARY

Notation	Definition
$\boldsymbol{x}, y$	Input $\boldsymbol{x} \in \mathbb{X}$ and one-hot label $y \in \mathbb{Y}$
$\mathbb{X}, \mathbb{Y} = [\![C]\!]$	Input and labels space
C	Output dimension: number of classes
$\mathcal{D} \subset \mathbb{X}$	Original training set sampled from p
$\mathcal{D}_r, \mathcal{D}_f$	Retain and forget set $\mathcal{D} = \mathcal{D}_r \cup \mathcal{D}_f$
$\mathcal{D}(y), \mathcal{D}_r(y), \mathcal{D}_f(y)$	Class-y subsets of the corresponding sets
$\lvert\mathcal{D}_f\rvert, \lvert\mathcal{D}_r\rvert, \lvert\mathcal{D}\rvert$	Sizes of sets $\mathcal{D}_f, \mathcal{D}_r, \mathcal{D}$
π_r, π_f	Mixture weights of the retain/forget components in p, $\pi_r + \pi_f = 1$
$s \in \{r, f\}$	State variable that doubles the labels number in the $2C$ construction (Sect. 4)
$\pi_r(y), \pi_f(y)$	Intra-class proportion $\lvert\mathcal{D}_s(y)\rvert / \lvert\mathcal{D}(y)\rvert$ used in the conditional mixture
p	True distribution of $(\boldsymbol{x}, y)$
$p_{\boldsymbol{x}}, p_y$	Marginal true distributions
$p(y \mid \boldsymbol{x}), p(\boldsymbol{x} \mid y)$	Conditional true distributions
p_r	True distribution of the retain set
p_f	True distribution of the forget set, $p = \pi_r p_r + \pi_f p_f$
$\boldsymbol{\ell}_{\boldsymbol{\theta}}(\boldsymbol{x})$	Logit vector output of the initial neural network at input $\boldsymbol{x}$, of parameters $\boldsymbol{\theta}$
$\boldsymbol{\ell}_{\boldsymbol{\theta}_u}(\boldsymbol{x})$	Logit vector of the distilled NN, of parameters $\boldsymbol{\theta}_u$
$f_{\boldsymbol{\theta}}(\boldsymbol{x}), f_{\boldsymbol{\theta}_r}(\boldsymbol{x})$	Predicted probabilities of the initial or retrained NN
$q_\eta(\boldsymbol{x})$	Target of the unlearned model: $q_\eta \doteq \mathrm{softmax}(\boldsymbol{\ell}_{\boldsymbol{\theta}} + \eta(\log \hat{p}_r - \log \hat{p}))$
$f_{\boldsymbol{\theta}_u}(\boldsymbol{x})$	Probability vectors of the unlearned NN (after distillation)
$\hat{p}_r(y \mid \boldsymbol{x}), \hat{p}(y \mid \boldsymbol{x})$	Posteriors induced by the proxies
$\hat{p}_r(\boldsymbol{x} \mid y), \hat{p}(\boldsymbol{x} \mid y)$	Proxy data models
$\hat{p}_f(\boldsymbol{x} \mid y)$	Per-class proxy of the forget distribution used in the conditional mixture
$\mathrm{softmax}$	Standard softmax operator
$\mathrm{LSE}(\boldsymbol{z})$	Log-Sum-Exponential: $\mathrm{LSE}(\boldsymbol{z}) \doteq \log \sum_{i=1}^{C} e^{z_i}$
$\gamma(\boldsymbol{x})$	Normalization constant of the initial NN: $\gamma(\boldsymbol{x}) \doteq \mathrm{LSE}(\boldsymbol{\ell}_{\boldsymbol{\theta}}(\boldsymbol{x}))$
$\zeta(\boldsymbol{x}; \eta)$	Normalization of the unlearned model: $\zeta(\boldsymbol{x}; \eta) \doteq \mathrm{LSE}(\log q_\eta(\boldsymbol{x}))$
$\mathrm{Z}(\boldsymbol{x}; \eta)$	Shift of the normalizations between the unlearned and the initial NN (23)
$\eta \geq 0$	Scaling parameter of the unlearning signal
$\eta_{\max}$	Largest $\eta > 0$ such that $\mathbb{E}\mathrm{Z}(\boldsymbol{x}; \eta) < 0$
$\mathbf{1}$	All-one vector in $\mathbb{R}^C$
$D_{\mathrm{KL}}^{y\mid\boldsymbol{x}}(p \,\|\, q)$	$\sum_{y=1}^{C} p(y \mid \boldsymbol{x}) \log \frac{p(y\mid\boldsymbol{x})}{q(y\mid\boldsymbol{x})}$. The conditional KL divergence
$\mathbb{E}D_{\mathrm{KL}}^{y\mid\boldsymbol{x}}(p \,\|\, q)$	Its expectation over $\boldsymbol{x} \sim p_{\boldsymbol{x}}$. Main figure of merit of the paper (Sect. 3.2)
$\mathbb{E}D_{\mathrm{KL}}^{\boldsymbol{x}\mid y}(p \,\|\, q)$	Symmetric expression conditioning on y in the bound of Prop. 3
$D_{\mathrm{KL}}^{\mathrm{t}}, D_{\mathrm{KL}}^{\mathrm{f}}$	Empirical expectation estimated on the test / forget set, as reported in the tables
$H(p)$	Shannon entropy of p

Table 3: List of the symbols with their short definition.

C JUSTIFICATIONS

We assume that the data are an independent sampling of the true distributions: $\mathcal{D} \sim p^{|\mathcal{D}|}$ and $\mathcal{D}_r \sim p_r^{|\mathcal{D}_r|}$. The original true distribution is a mixture of form (Le et al., 2026):

$$p = \pi_r p_r + \pi_f p_f \quad \text{with} \quad \pi_r + \pi_f = 1. \tag{14}$$

C.1 THE UNLEARNED MODEL FAMILY

We look for a model q, which is an update similar to the initial NN $f_{\boldsymbol{\theta}}$, closer to the ideal retain distribution p_r than the initial distribution p. This objective is framed in terms of KL divergences in the following optimization problem:

$$\min_q \ \mathbb{E}D_{\mathrm{KL}}^{y\mid\boldsymbol{x}}(q \,\|\, f_{\boldsymbol{\theta}}) \quad \text{s.t.} \quad \mathbb{E}D_{\mathrm{KL}}^{y\mid\boldsymbol{x}}(q \,\|\, p_r) \leq \mathbb{E}D_{\mathrm{KL}}^{y\mid\boldsymbol{x}}(q \,\|\, p), \quad \mathbb{E}\sum_y q(y \mid \boldsymbol{x}) = 1. \tag{15}$$

By definition of the D_{KL}, one always has the identity

$$\mathbb{E}D_{\mathrm{KL}}^{y|\boldsymbol{x}}(q \,\|\, p) - \mathbb{E}D_{\mathrm{KL}}^{y|\boldsymbol{x}}(q \,\|\, p_r) = \mathbb{E}\sum_y q\left(y \mid \boldsymbol{x}\right) \log \frac{p_r(y \mid \boldsymbol{x})}{p\left(y \mid \boldsymbol{x}\right)}.$$

Following (Cover and Thomas, 2006, Chap. 12), we write the Lagrangian formulation of problem (15) with multipliers $\eta \geq 0$ and ν:

$$J(q; \eta, \nu) \doteq \mathbb{E}\sum_y q\left(y \mid \boldsymbol{x}\right) \log \frac{q\left(y \mid \boldsymbol{x}\right)}{f_{\boldsymbol{\theta}}\left(y \mid \boldsymbol{x}\right)} - \eta\mathbb{E}\sum_y q\left(y \mid \boldsymbol{x}\right) \log \frac{p_r(y \mid \boldsymbol{x})}{p\left(y \mid \boldsymbol{x}\right)} + \nu\mathbb{E}\sum_y q\left(y \mid \boldsymbol{x}\right).$$

Cancelling the derivative with respect to $q\left(y \mid \boldsymbol{x}\right)$, we obtain: $\forall y \in \mathbb{Y}$

$$\mathbb{E}\log \frac{q\left(y \mid \boldsymbol{x}\right)}{f_{\boldsymbol{\theta}}\left(y \mid \boldsymbol{x}\right)} + 1 - \eta\mathbb{E}\log \frac{p_r(y \mid \boldsymbol{x})}{p\left(y \mid \boldsymbol{x}\right)} + \nu = 0. \tag{16}$$

Rearranging the terms, the final expression can be expressed as:

$$\mathbb{E}\left[\log(q_\eta(\cdot \mid \boldsymbol{x}))\right] = \mathbb{E}\left[\underbrace{\log f_{\boldsymbol{\theta}}\left(\cdot \mid \boldsymbol{x}\right)}_{\boldsymbol{\ell}_{\boldsymbol{\theta}}(\boldsymbol{x}) - \gamma(\boldsymbol{x})\mathbf{1}} + \eta \underbrace{\log \frac{p_r(\cdot \mid \boldsymbol{x})}{p\left(\cdot \mid \boldsymbol{x}\right)}}_{\text{exact unlearning signal } \log p_r(\cdot|\boldsymbol{x}) - \log p(\cdot|\boldsymbol{x})} - \underbrace{\mathrm{Z}(\boldsymbol{x}; \eta)}_{\text{normalizing term}}\right]. \tag{17}$$

Remark. In (Cover and Thomas, 2006, Chap. 12), $\eta = \eta_0$ is chosen so that:

$$\mathbb{E}D_{\mathrm{KL}}^{y|\boldsymbol{x}}(q_{\eta_0} \,\|\, p) - \mathbb{E}D_{\mathrm{KL}}^{y|\boldsymbol{x}}(q_{\eta_0} \,\|\, p_r) = 0.$$

An attacker observing samples from q is not, in expectation, in capacity to decide whether the drawn samples are coming from p_r or p.

Estimation. One solution is to impose that (17) holds for any $\boldsymbol{x}$. This translates in the logit space to the solution:

$$\boldsymbol{\ell}_{\boldsymbol{\theta}}(\boldsymbol{x}) + \eta(\log p_r(\cdot \mid \boldsymbol{x}) - \log p(\cdot \mid \boldsymbol{x})),$$

i.e. the family of unlearned model presented in Sect. 3. The true exact unlearning signal $\log p_r(\cdot \mid \boldsymbol{x}) - \log p(\cdot \mid \boldsymbol{x})$ being inaccessible in practice, our key idea consists in relying on proxies $(\hat{p}, \hat{p}_r)$:

$$\log \hat{p}_r(\cdot \mid \boldsymbol{x}) - \log \hat{p}(\cdot \mid \boldsymbol{x}) \approx \log p_r(\cdot \mid \boldsymbol{x}) - \log p(\cdot \mid \boldsymbol{x}).$$

Rephrasing, given proxies $(\hat{p}, \hat{p}_r)$ we obtain our unlearned model family of type

$$\boldsymbol{\ell}_{\boldsymbol{\theta}}(\boldsymbol{x}) + \eta(\log \hat{p}_r(\cdot \mid \boldsymbol{x}) - \log \hat{p}(\cdot \mid \boldsymbol{x})).$$

More precisely, it is the unbiased solution of the following optimization problem:

$$\min_q \ \mathbb{E}D_{\mathrm{KL}}^{y|\boldsymbol{x}}(q \,\|\, f_{\boldsymbol{\theta}}) \quad \text{s.t.} \quad \mathbb{E}D_{\mathrm{KL}}^{y|\boldsymbol{x}}(q \,\|\, \hat{p}_r) \leq \mathbb{E}D_{\mathrm{KL}}^{y|\boldsymbol{x}}(q \,\|\, \hat{p}), \quad \mathbb{E}\sum_y q\left(y \mid \boldsymbol{x}\right) = 1.$$

To conclude, we consider the following family of unlearned models:

$$\log q_\eta(\cdot \mid \boldsymbol{x}) \doteq \boldsymbol{\ell}_{\boldsymbol{\theta}}(\boldsymbol{x}) + \eta\left(\log \hat{p}_r - \log \hat{p}\right)(\cdot \mid \boldsymbol{x}) - \zeta(\boldsymbol{x}; \eta)\mathbf{1}, \tag{18}$$

for some normalization constant $\zeta(\boldsymbol{x}; \eta)$. The logits and probits are related by:

$$q_\eta(\boldsymbol{x}) \doteq \mathrm{softmax}\left(\log q_\eta(\boldsymbol{x})\right) = \mathrm{softmax}\left(\boldsymbol{\ell}_{\boldsymbol{\theta}}(\boldsymbol{x}) + \eta\left(\log \hat{p}_r - \log \hat{p}\right)(\cdot \mid \boldsymbol{x})\right). \tag{19}$$

C.2 THE SAFETY RANGE OF η

We first relate the normalization $\zeta(\boldsymbol{x};\eta)$ to $\gamma(\boldsymbol{x})$, the normalization of the initial NN.

$\mathrm{LSE}:\mathbb{R}^C\to\mathbb{R}$ denotes the Log-Sum-Exponential function defined as $\mathrm{LSE}(\boldsymbol{z})\doteq\log\left(\sum_{i=1}^C e^{z_i}\right)$. It verifies, $\forall\boldsymbol{z}\in\mathbb{R}^C,\forall\kappa\in\mathbb{R}$:

$$\log(\mathrm{softmax}(\boldsymbol{z}))=\boldsymbol{z}-\mathrm{LSE}(\boldsymbol{z})\mathbf{1} \tag{20}$$

$$\mathrm{LSE}(\boldsymbol{z}+\kappa\mathbf{1})=\mathrm{LSE}(\boldsymbol{z})+\kappa \tag{21}$$

$$\mathrm{LSE}(\log(\boldsymbol{z}))=\log\left(\sum_{i=1}^C z_i\right), \tag{22}$$

where $\log(\boldsymbol{z})$ is an abuse of notation defining the vector $(\log z_1,\dots,\log z_C)$ for $\boldsymbol{z}\in\mathbb{R}^C_{>0}$, and $\mathbf{1}\in\mathbb{R}^C$ is the all-one vector.

Indeed, applying LSE to the logits of the unlearned model $\log q_\eta(\boldsymbol{x})=\log q_\eta(\boldsymbol{x})+\zeta(\boldsymbol{x};\eta)\mathbf{1}$, isolates the normalization term due to (21) and (22): $\mathrm{LSE}(\log q_\eta(\boldsymbol{x}))=\zeta(\boldsymbol{x};\eta)$. Since we forge the unlearned model by adding the proxy shift to the initial NN (3), we have:

$$\begin{aligned}\zeta(\boldsymbol{x};\eta)&\overset{(18)}{=}\mathrm{LSE}\left(\boldsymbol{\ell}_{\boldsymbol{\theta}}(\boldsymbol{x})+\eta\log\left(\frac{\hat{p}_r(\cdot\mid\boldsymbol{x})}{\hat{p}\,(\cdot\mid\boldsymbol{x})}\right)\right)\\&\overset{(19)}{=}\mathrm{LSE}\left(\log f_{\boldsymbol{\theta}}(\boldsymbol{x})+\gamma(\boldsymbol{x})\mathbf{1}+\eta\log\left(\frac{\hat{p}_r(\cdot\mid\boldsymbol{x})}{\hat{p}\,(\cdot\mid\boldsymbol{x})}\right)\right)\\&\overset{(21)}{=}\gamma(\boldsymbol{x})+\mathrm{LSE}\left(\log\left(\frac{f_{\boldsymbol{\theta}}(\boldsymbol{x})\hat{p}_r(\cdot\mid\boldsymbol{x})^\eta}{\hat{p}^\eta\,(\cdot\mid\boldsymbol{x})}\right)\right),\end{aligned}$$

with component-wise multiplication and division between vectors. This links the normalization offset ζ of the unlearned model to γ, the one of the initial model. Thanks to (22), we have:

$$\mathrm{Z}(\boldsymbol{x};\eta)\doteq\zeta(\boldsymbol{x};\eta)-\gamma(\boldsymbol{x})=\log\left(\sum_{y=1}^C f_{\boldsymbol{\theta}}\,(y\mid\boldsymbol{x})\left(\frac{\hat{p}_r(y\mid\boldsymbol{x})}{\hat{p}\,(y\mid\boldsymbol{x})}\right)^\eta\right) \tag{23}$$

Note that $\mathrm{Z}(\boldsymbol{x};0)=0$, $\forall\boldsymbol{x}$, which is expected as the unlearning operator is void at $\eta=0$. Its derivative w.r.t. η equals

$$\frac{\partial\mathrm{Z}(\boldsymbol{x};\eta)}{\partial\eta}=\frac{\mathbb{E}_{y\sim f_{\boldsymbol{\theta}}(y|\boldsymbol{x})}\left[A^\eta(y,\boldsymbol{x})\log\left(A(y,\boldsymbol{x})\right)\right]}{\mathbb{E}_{y\sim f_{\boldsymbol{\theta}}(y|\boldsymbol{x})}\left[A^\eta(y,\boldsymbol{x})\right]},$$

with $A(y,\boldsymbol{x})\doteq{}^{\hat{p}_r(y|\boldsymbol{x})}\!/_{\hat{p}(y|\boldsymbol{x})}$, and because the derivative of $a^u=\exp(u\log(a))$ equals $a^u\log(a)$, $\forall a>0$. Note that, at point $\eta=0$,

$$\begin{aligned}\left.\frac{\partial\mathrm{Z}(\boldsymbol{x};\eta)}{\partial\eta}\right|_{\eta=0}&=\mathbb{E}_{y\sim f_{\boldsymbol{\theta}}(y|\boldsymbol{x})}\left[\log\left(\frac{\hat{p}_r(y\mid\boldsymbol{x})}{\hat{p}\,(y\mid\boldsymbol{x})}\right)\right]\\&=D_{\mathrm{KL}}^{y|\boldsymbol{x}}(f_{\boldsymbol{\theta}}\,\|\,\hat{p})-D_{\mathrm{KL}}^{y|\boldsymbol{x}}(f_{\boldsymbol{\theta}}\,\|\,\hat{p}_r)\,.\end{aligned}$$

On expectation over $\boldsymbol{x}\sim p_{\boldsymbol{x}}$ we obtain:

$$\mathbb{E}\left[\left.\frac{\partial\mathrm{Z}(\boldsymbol{x};\eta)}{\partial\eta}\right|_{\eta=0}\right]=\mathbb{E}D_{\mathrm{KL}}^{y|\boldsymbol{x}}(p\,\|\,\hat{p})-\mathbb{E}D_{\mathrm{KL}}^{y|\boldsymbol{x}}(p\,\|\,\hat{p}_r)\,.$$

This quantity is negative if $\hat{p}$ is a better proxy of p than $\hat{p}_r$.

In the same way, the second derivative verifies, $\forall\boldsymbol{x}\in\mathbb{X}$:

$$\frac{\partial^2\mathrm{Z}(\boldsymbol{x};\eta)}{\partial^2\eta}=\frac{\mathbb{E}_{y\sim f_{\boldsymbol{\theta}}(y|\boldsymbol{x})}\left[A^\eta(y,\boldsymbol{x})\log^2\left(A(y,\boldsymbol{x})\right)\right]}{\mathbb{E}_{y\sim f_{\boldsymbol{\theta}}(y|\boldsymbol{x})}\left[A^\eta(y,\boldsymbol{x})\right]}-\frac{\mathbb{E}_{y\sim f_{\boldsymbol{\theta}}(y|\boldsymbol{x})}\left[A^\eta(y,\boldsymbol{x})\log\left(A(y,\boldsymbol{x})\right)\right]^2}{\left(\mathbb{E}_{y\sim f_{\boldsymbol{\theta}}(y|\boldsymbol{x})}\left[A^\eta(y,\boldsymbol{x})\right]\right)^2}\geq 0,$$

thanks to the Cauchy-Schwarz inequality.

Finally, Z is coercive for non zero logits shift because, in this case, at least one of the coordinates y verifies $\frac{\hat{p}_r(y|\boldsymbol{x})}{\hat{p}(y|\boldsymbol{x})}>1$.

To recap, $\mathrm{Z}(\eta)\doteq\mathbb{E}\left[\mathrm{Z}(\boldsymbol{x};\eta)\right]$ enjoys the following properties:

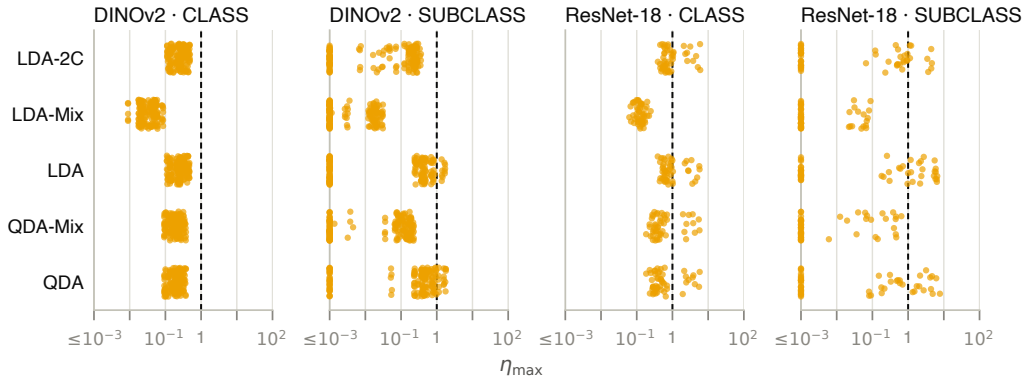


Figure 6: Computation of $\eta_{\max}$ from the proxies, over scenarios and architecture. One dot corresponds to one run of Alg. 2 across seeds, specific class and subclass scenario. Results for the three heads of DINOv2 are aggregated onto one subfigure.

- $\mathrm{Z}(0) = 0$
- $\mathrm{Z}(\eta)$ is convex over $[0, +\infty)$
- Z is coercive for non zero logits shift
- $\mathrm{Z}'(0) \leq 0$ if $\hat{p}$ is a better proxy of p than $\hat{p}_r$.

Consequently, under this assumption, there exists $\eta_{\max} > 0$ such that $\mathrm{Z}(\eta) \leq 0$ for all $0 \leq \eta \leq \eta_{\max}$. If the assumption does not hold, then $\eta_{\max} = 0$ and there is no unlearning.

In practice, $\eta_{\max}$ is found by a line search as the zero of the convex Monte-Carlo approximation:

$$\hat{Z}(\eta) \doteq \frac{1}{N} \sum_{\boldsymbol{x} \in \mathcal{D}} \mathrm{LSE}(\boldsymbol{\ell}_{\boldsymbol{\theta}}(\boldsymbol{x}) + \eta \, (\log \hat{p}_r - \log \hat{p})(\boldsymbol{x})) - \mathrm{LSE}(\boldsymbol{\ell}_{\boldsymbol{\theta}}(\boldsymbol{x})) \simeq \mathrm{Z}(\eta). \tag{24}$$

This is summarized in Alg. 2.

Remark on the interpretation of $\eta_{\max}$. The careful reader will spot that the derivative of Z cancels out exactly at η_0: the regime of indistinguishability between $\hat{p}$ and $\hat{p}_r$. We have $\eta_0 \leq \eta_{\max}$ due to the negative derivative hypothesis. Taking $\eta = \eta_{\max}$ ensures that we have been through the equality boundary and that we are in the regime where $\mathbb{E} D_{\mathrm{KL}}^{y|\boldsymbol{x}}(q_\eta \,\|\, \hat{p}) > \mathbb{E} D_{\mathrm{KL}}^{y|\boldsymbol{x}}(q_\eta \,\|\, \hat{p}_r)$.

Remark on the algorithmic convention. Alg 2 needs an upper bound η_{up} for initializing the search. If $\hat{Z}(\eta_{up}) \leq 0$ returning $\eta_{\max} = \eta_{up}$ remains safe. But the true zero may lie beyond η_{up}. Then, the bisection might be preceded by a phase that geometrically increases the upper bound until $\hat{Z}(\eta_{\mathrm{up}}) > 0$. This corresponds to the classical exponential search for unbounded problems (Bentley and Yao, 1976). Section 5 sets $\eta_{up} = 100$ to avoid this algorithmic complexity. The algorithmic choice was largely sufficient for our setup (see Fig. 6).

D Main results

D.1 Study of the logits error and proof of Prop. 1

We now define the error of prediction between the true distribution $p_r\,(y \mid \boldsymbol{x})$ and the unlearned model q_η, in log scale:

$$\mathcal{E}(y, \boldsymbol{x}; \eta) \doteq \log(p_r(y \mid \boldsymbol{x})) - \log(q_\eta(y \mid \boldsymbol{x})).$$

Algorithm 2 ETAMAX — the safety-range line search.

Require: logits $\boldsymbol{\ell}_{\boldsymbol{\theta}}(\boldsymbol{x})$, log ratios $\hat{p}_r(y|\boldsymbol{x})/\hat{p}(y|\boldsymbol{x})$, upper search η_{up}, $N = |\mathcal{D}|$, tolerance τ.
Ensure: Safety range $\eta_{\max}$ of the unlearned family (3) with $Z(\eta_{\max}) \leq 0$.

function $\hat{Z}(\eta)$ ▷ the empirical Z of (24)
  return $\frac{1}{N}\sum_{\boldsymbol{x}\in\mathcal{D}}\big[\text{LSE}\big(\boldsymbol{\ell}_{\boldsymbol{\theta}}(\boldsymbol{x}) + \eta\,(\log\hat{p}_r - \log\hat{p})(\boldsymbol{x})\big) - \text{LSE}\big(\boldsymbol{\ell}_{\boldsymbol{\theta}}(\boldsymbol{x})\big)\big]$
end function
$\hat{Z}'(0) \leftarrow \frac{1}{N}\sum_{\boldsymbol{x}\in\mathcal{D}}\sum_{y=1}^{C} f_{\boldsymbol{\theta}}(\boldsymbol{x})_y \log\big(\hat{p}_r(y|\boldsymbol{x})/\hat{p}(y|\boldsymbol{x})\big)$
if $\hat{Z}'(0) \geq 0$ **then**
  return 0 ▷ safe if not decreasing
end if
$\eta_{\text{down}} \leftarrow 0$
if $\hat{Z}(\eta_{\text{up}}) \leq 0$ **then**
  return η_{up}
end if
while $\eta_{\text{up}} - \eta_{\text{down}} > \tau$ **do** ▷ invariant: $\hat{Z}(\eta_{\text{down}}) \leq 0 < \hat{Z}(\eta_{\text{up}})$
  $\eta \leftarrow (\eta_{\text{down}} + \eta_{\text{up}})/2$
  if $\hat{Z}(\eta) \leq 0$ **then** $\eta_{\text{down}} \leftarrow \eta$ **else** $\eta_{\text{up}} \leftarrow \eta$ ▷ convexity makes the zero unique
end while
return η_{down} ▷ conservative end so that $\hat{Z}(\eta_{\max}) \leq 0$ holds

This quantity is worth studying as it appears in the computation of $\mathbb{E}D_{\text{KL}}^{y|\boldsymbol{x}}(p_r \,\|\, q_\eta)$, the quantity of importance in this paper. We have:

$$D_{\text{KL}}^{y|\boldsymbol{x}}(p_r \,\|\, q_\eta) = \sum_{y=1}^{C} p_r(y \mid \boldsymbol{x}) \log\left(\frac{p_r(y \mid \boldsymbol{x})}{q_\eta(y \mid \boldsymbol{x})}\right) = \langle p_r(\cdot \mid \boldsymbol{x}), \mathcal{E}(\cdot, \boldsymbol{x}, \eta)\rangle. \tag{25}$$

On the other hand, we have

$$\begin{aligned}
\mathcal{E}(\boldsymbol{x}, y; \eta) &= \log(p_r(y \mid \boldsymbol{x})) - \log q_\eta(\boldsymbol{x}) + \zeta(\boldsymbol{x}; \eta) \\
&\overset{(3)}{=} \log(p_r(y \mid \boldsymbol{x})) - \boldsymbol{\ell}_{\boldsymbol{\theta}}(\boldsymbol{x}) - \eta(\log\hat{p}_r - \log\hat{p})(\boldsymbol{x}) + \zeta(\boldsymbol{x}; \eta) \\
&= \log(p_r(y \mid \boldsymbol{x})) - \log f_{\boldsymbol{\theta}}(y \mid \boldsymbol{x}) - \gamma(\boldsymbol{x}) - \eta(\log\hat{p}_r - \log\hat{p})(\boldsymbol{x}) + \zeta(\boldsymbol{x}; \eta) \\
&= \log\left(\frac{p_r(y \mid \boldsymbol{x})}{f_{\boldsymbol{\theta}}(y \mid \boldsymbol{x})}\right) - \eta \log\left(\frac{\hat{p}_r(y \mid \boldsymbol{x})}{\hat{p}(y \mid \boldsymbol{x})}\right) + \text{Z}(\boldsymbol{x}; \eta)
\end{aligned} \tag{26}$$

Plugging (26) into (25) and taking the expectation leads to:

$$\begin{aligned}
\mathbb{E}D_{\text{KL}}^{y|\boldsymbol{x}}(p_r \,\|\, q_\eta) = \mathbb{E}D_{\text{KL}}^{y|\boldsymbol{x}}(p_r \,\|\, f_{\boldsymbol{\theta}}) \\
+ \underbrace{\mathbb{E}\left[\text{Z}(\boldsymbol{x}, \eta)\right]}_{\leq 0, \forall \eta \in [0, \eta_{\max}]} + \eta \left(\underbrace{\mathbb{E}D_{\text{KL}}^{y|\boldsymbol{x}}(p_r \,\|\, \hat{p}_r) - \mathbb{E}D_{\text{KL}}^{y|\boldsymbol{x}}(p_r \,\|\, \hat{p})}_{\leq 0 \text{ if } \hat{p}_r \text{ is a better proxy for } p_r \text{ than } \hat{p} \text{ is}}\right).
\end{aligned}$$

The first term is the KL divergence without unlearning, *i.e.* between the true retain distribution and the initial NN probits. App. C.2 shows that the second term is negative for any $0 \leq \eta \leq \eta_{\max}$. The third term is negative with one extra assumption: if $\hat{p}_r$ is a better proxy for p_r than $\hat{p}$ is.

This proves Prop. 1.

D.2 PROOF OF PROP. 2

We rewrite (26) in order to obtain another expression of the error $\mathcal{E}$:

$$\mathcal{E}(\boldsymbol{x}, y; \eta) = (1 - \eta) \log\left(\frac{p_r(y \mid \boldsymbol{x})}{f_{\boldsymbol{\theta}}(y \mid \boldsymbol{x})}\right) + \eta \left(\log\left(\frac{p_r(y \mid \boldsymbol{x})}{\hat{p}_r(y \mid \boldsymbol{x})}\right) - \log\left(\frac{f_{\boldsymbol{\theta}}(y \mid \boldsymbol{x})}{\hat{p}(y \mid \boldsymbol{x})}\right)\right) + \text{Z}(\boldsymbol{x}; \eta).$$

We now take the expectation w.r.t. $p_r\left(y \mid \boldsymbol{x}\right)$ and then the expectation w.r.t. $p_{\boldsymbol{x}}$. For $\eta \in [0, \eta_{\max}]$, $\mathrm{Z}(\eta)$ is non-positive. This leads to:

$$\mathbb{E}D_{\mathrm{KL}}^{y|\boldsymbol{x}}(p_r \,\|\, q_\eta) \leq (1-\eta)\mathbb{E}D_{\mathrm{KL}}^{y|\boldsymbol{x}}(p_r \,\|\, f_{\boldsymbol{\theta}}) + \eta \left(\mathbb{E}D_{\mathrm{KL}}^{y|\boldsymbol{x}}(p_r \,\|\, \hat{p}_r) - \underbrace{\mathbb{E}\sum_{y=1}^{C} p_r(y \mid \boldsymbol{x}) \log \frac{f_{\boldsymbol{\theta}}\left(y \mid \boldsymbol{x}\right)}{\hat{p}\left(y \mid \boldsymbol{x}\right)}}_{\text{term } A} \right)$$

Note that we get an equality when $\eta = \eta_{\max}$ because $\mathrm{Z}(\eta_{\max}) = 0$ by definition of $\eta_{\max}$. The following lemma helps us work out term A.

Lemma 1 (Formal Calculus)**.** *According to the mixture* (14) *of the true distributions, $p = (1 - \pi_f)p_r + \pi_f p_f$ over the joint data $(\boldsymbol{x}, y)$. Then we have the following formal identity with respect to $\boldsymbol{x}$, y, or $(\boldsymbol{x}, y)$:*

$$\mathbb{E} = (1-\pi_f)\mathbb{E}_r + \pi_f \mathbb{E}_f = \mathbb{E}_r + \pi_f \left(\mathbb{E}_f - \mathbb{E}_r\right) = \mathbb{E}_r + \mathcal{O}(\pi_f). \tag{27}$$

Applying this lemma to the term $A = \mathbb{E}\left[\mathrm{z}\right]$, where we abbreviate $\mathrm{z} \doteq \sum_{y=1}^{C} p_r(y \mid \boldsymbol{x}) \log \frac{f_{\boldsymbol{\theta}}(y|\boldsymbol{x})}{\hat{p}(y|\boldsymbol{x})}$:

$$A \overset{(27)}{=} \mathbb{E}_r[\mathrm{z}] + \pi_f \left(\mathbb{E}_f\left[\mathrm{z}\right] - \mathbb{E}_r[\mathrm{z}]\right).$$

Lemma 2 (Expectation Telescoping)**.** *For a measure μ over $(\boldsymbol{x}, y)$ and any test function $\phi(\boldsymbol{x}, y)$ we have the following identity:*

$$\mathbb{E}_\mu\left[\phi(\boldsymbol{x}, y)\right] = \mathbb{E}_\mu\left[\sum_{y=1}^{C} \mu(y \mid \boldsymbol{x})\phi(\boldsymbol{x}, y)\right]. \tag{28}$$

On the right-hand side of the equation, the expectation is taken on a quantity that do not depend on y.

We note $\Lambda(\boldsymbol{x}, y) \doteq \log \frac{f_{\boldsymbol{\theta}}(y|\boldsymbol{x})}{\hat{p}(y|\boldsymbol{x})}$ and use lemma 2 to write $\mathbb{E}_{\boldsymbol{x}\sim p_r}[\mathrm{z}] = \mathbb{E}_{(\boldsymbol{x},y)\sim p_r}[\Lambda(\boldsymbol{x}, y)]$.

Expanding $\mathbb{E}_r[\mathrm{z}]$ with the same lemma, now applied on the joint $(\boldsymbol{x}, y)$,

$$\begin{aligned}
A &\overset{(27)}{=} \mathbb{E}_r[\mathrm{z}] + \pi_f \left(\mathbb{E}_f\left[\mathrm{z}\right] - \mathbb{E}_r[\mathrm{z}]\right) \\
&\overset{(28)}{=} \mathbb{E}_r[\Lambda(\boldsymbol{x}, y)] + \pi_f \left(\mathbb{E}_f\left[\mathrm{z}\right] - \mathbb{E}_r[\mathrm{z}]\right) \\
&\overset{(27)}{=} \mathbb{E}[\Lambda(\boldsymbol{x}, y)] - \pi_f(\mathbb{E}_f[\Lambda(\boldsymbol{x}, y)] - \mathbb{E}_r[\Lambda(\boldsymbol{x}, y)]) + \pi_f \left(\mathbb{E}_f\left[\mathrm{z}\right] - \mathbb{E}_r[\mathrm{z}]\right) \\
&\overset{(28)}{=} \mathbb{E}[\Lambda(\boldsymbol{x}, y)] - \pi_f \mathbb{E}_f[\Lambda(\boldsymbol{x}, y)] + \pi_f \mathbb{E}_f\left[\mathrm{z}\right] \\
&= \underbrace{\mathbb{E}\left[\Lambda(\boldsymbol{x}, y)\right]}_{\text{term } A_1} + \underbrace{\pi_f \mathbb{E}_f\left[\sum_{y=1}^{C} ((p_r(y \mid \boldsymbol{x}) - p_f(y \mid \boldsymbol{x}))\Lambda(\boldsymbol{x}, y))\right]}_{\text{term } A_2},
\end{aligned}$$

where the last line uses $\mathbb{E}_f[\Lambda(\boldsymbol{x}, y)] \overset{(28)}{=} \mathbb{E}_f\left[\sum_y p_f(y \mid \boldsymbol{x})\, \Lambda(\boldsymbol{x}, y)\right]$.

The first term A_1 is $\mathbb{E}\left[\Lambda(\boldsymbol{x}, y)\right] = \mathbb{E}\left[\log \frac{f_{\boldsymbol{\theta}}(y|\boldsymbol{x})}{\hat{p}(y|\boldsymbol{x})}\right]$. Here, we separate the unlearning error from the learning error $\epsilon_{\texttt{training}}$:

$$\epsilon_{\texttt{training}} \doteq \mathbb{E}D_{\mathrm{KL}}^{y|\boldsymbol{x}}(p \,\|\, f_{\boldsymbol{\theta}}) = \mathbb{E}\left[\log \frac{p\left(y \mid \boldsymbol{x}\right)}{f_{\boldsymbol{\theta}}\left(y \mid \boldsymbol{x}\right)}\right] \tag{29}$$

so that we can state the bound in terms of divergence using (28) with

$$A_1 = \mathbb{E}D_{\mathrm{KL}}^{y|\boldsymbol{x}}(p \,\|\, \hat{p}) - \mathbb{E}D_{\mathrm{KL}}^{y|\boldsymbol{x}}(p \,\|\, f_{\boldsymbol{\theta}}). \tag{30}$$

The second term A_2 is not really tractable and will be referred to as a residual $\mathcal{O}(\pi_f)$:

$$A_2 = \pi_f \mathbb{E}_f\left[\sum_{y=1}^{C} \left((p_r(y \mid \boldsymbol{x}) - p_f(y \mid \boldsymbol{x})) \log \frac{f_{\boldsymbol{\theta}}\left(y \mid \boldsymbol{x}\right)}{\hat{p}\left(y \mid \boldsymbol{x}\right)}\right)\right] = \mathcal{O}(\pi_f). \tag{31}$$

The residual is close to zero either when $p_f(y \mid \boldsymbol{x})$ and $p_r(y \mid \boldsymbol{x})$ take similar values on most likely forget data points (no unlearning signal), or when the initial proxy is close to the initial distribution.

In the end, since A enters the bound with a minus sign and a factor η, we obtain $\mathbb{E}D_{\mathrm{KL}}^{y|\boldsymbol{x}}(p_r \,\|\, q_\eta)$ lower or equal to

$$(1-\eta)\mathbb{E}D_{\mathrm{KL}}^{y|\boldsymbol{x}}(p_r \,\|\, f_{\boldsymbol{\theta}}) + \eta\left(\mathbb{E}D_{\mathrm{KL}}^{y|\boldsymbol{x}}(p_r \,\|\, \hat{p}_r) - \mathbb{E}D_{\mathrm{KL}}^{y|\boldsymbol{x}}(p \,\|\, \hat{p})\right) - \eta\,\mathcal{O}(\pi_f) + \eta\epsilon_{\texttt{training}}. \quad (32)$$

This proves Prop. 2.

D.3 Proof of Prop. 3

This section converts the previous bound with probabilities $y \mid \boldsymbol{x}$ into terms modeling $\boldsymbol{x} \mid y$ thanks to the following lemma.

Lemma 3. *For any distributions P and Q over $(\boldsymbol{x}, y)$ with the same marginal $P(y) = Q(y)$:*

$$\mathbb{E}\left[D_{\mathrm{KL}}^{y|\boldsymbol{x}}(P \,\|\, Q)\right] = \mathbb{E}\left[D_{\mathrm{KL}}^{\boldsymbol{x}|y}(P \,\|\, Q)\right] - D_{\mathrm{KL}}(P_{\boldsymbol{x}} \,\|\, Q_{\boldsymbol{x}}) \quad (33)$$

Proof.

$$\begin{aligned}
\mathbb{E}\left[D_{\mathrm{KL}}^{y|\boldsymbol{x}}(P \,\|\, Q)\right] &= \mathbb{E}_{\boldsymbol{x}\sim P}\left[D_{\mathrm{KL}}^{y|\boldsymbol{x}}(P \,\|\, Q)\right] \\
&= \mathbb{E}_{\boldsymbol{x}\sim P}\left[\mathbb{E}_{y\sim P(y|\boldsymbol{x})}\left[\log \frac{P\,(y \mid \boldsymbol{x})}{Q\,(y \mid \boldsymbol{x})}\right]\right] \\
&= \mathbb{E}_{(\boldsymbol{x},y)\sim P}\left[\log \frac{P\,(y \mid \boldsymbol{x})}{Q\,(y \mid \boldsymbol{x})}\right] \\
&\overset{\text{Bayes}}{=} \mathbb{E}_{(\boldsymbol{x},y)\sim P}\left[\log \frac{P\,(\boldsymbol{x} \mid y)\,Q(\boldsymbol{x})}{Q\,(\boldsymbol{x} \mid y)\,P(\boldsymbol{x})}\right] \\
&= \mathbb{E}_{y\sim P}\left[D_{\mathrm{KL}}^{\boldsymbol{x}|y}(P \,\|\, Q)\right] - D_{\mathrm{KL}}(P_{\boldsymbol{x}} \,\|\, Q_{\boldsymbol{x}}) \\
&= \mathbb{E}\left[D_{\mathrm{KL}}^{\boldsymbol{x}|y}(P \,\|\, Q)\right] - D_{\mathrm{KL}}(P_{\boldsymbol{x}} \,\|\, Q_{\boldsymbol{x}})\,,
\end{aligned}$$

because $\mathbb{E}_{(\boldsymbol{x},y)\sim P}\left[\log \frac{P(\boldsymbol{x})}{Q(\boldsymbol{x})}\right] = \mathbb{E}_{\boldsymbol{x}\sim P}\left[\log \frac{P(\boldsymbol{x})}{Q(\boldsymbol{x})}\right] = D_{\mathrm{KL}}(P_{\boldsymbol{x}} \,\|\, Q_{\boldsymbol{x}})$ considers only the marginal distributions of $\boldsymbol{x}$. □

By definition of our proxies they have the same marginal in y than the distribution they are attributed to. Lemma 3 is legitimate on both lines: $\hat{p}_{r,y} = p_{r,y}$ and $\hat{p}_y = p_y$ hold by construction of the proxy posteriors (8), which reuse the true class priors. Applying Lemma 3 to the second term of the previous bound leads to:

$$\begin{aligned}
\mathbb{E}D_{\mathrm{KL}}^{y|\boldsymbol{x}}(p_r \,\|\, \hat{p}_r) - \mathbb{E}D_{\mathrm{KL}}^{y|\boldsymbol{x}}(p \,\|\, \hat{p}) &\overset{(27)}{=} \mathbb{E}_r D_{\mathrm{KL}}^{y|\boldsymbol{x}}(p_r \,\|\, \hat{p}_r) - \mathbb{E}D_{\mathrm{KL}}^{y|\boldsymbol{x}}(p \,\|\, \hat{p}) + \mathcal{O}(\pi_f) \\
&\overset{(33)}{=} \mathbb{E}_r D_{\mathrm{KL}}^{\boldsymbol{x}|y}(p_r \,\|\, \hat{p}_r) - D_{\mathrm{KL}}(p_{r,\boldsymbol{x}} \,\|\, \hat{p}_{r,\boldsymbol{x}}) \\
&\quad - \mathbb{E}D_{\mathrm{KL}}^{\boldsymbol{x}|y}(p \,\|\, \hat{p}) + D_{\mathrm{KL}}(p_{\boldsymbol{x}} \,\|\, \hat{p}_{\boldsymbol{x}}) + \mathcal{O}(\pi_f).
\end{aligned}$$

This proves Prop. 3.

E About discriminant analysis proxies

In the experiments Sect. 5 the reported QDA are taken with diagonal covariances for sake of computational cost. It explains why they can under-perform our LDA approaches. This section derives the cost of not taking the complete covariances in our QDA experiments, linking it directly to the performances of our LDA methods (see Prop. 4).

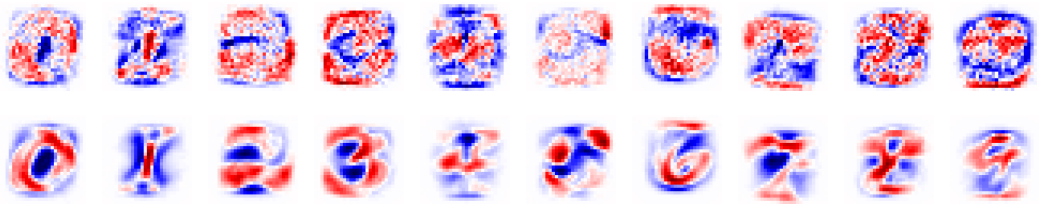


Figure 7: LDA **matches cross-entropy training weights.** Top: last-layer weight matrix (rows reshaped as images) of a linear network trained with the Adam optimizer and cross-entropy loss. Bottom: closed-form LDA weight formula. Dataset: MNIST (LeCun et al., 1998). The near-perfect visual match empirically confirms the cross-entropy/LDA equivalence.

Homoscedastic Gaussian data. Linear Discriminant Analysis (LDA) assumes a generative model where each class follows a Gaussian distribution with a shared covariance:

$$\boldsymbol{x} \mid y = k \;\sim\; \mathcal{N}(\boldsymbol{\mu}_k,\, \boldsymbol{\Sigma}). \tag{34}$$

The posterior probability has the following expression (McLachlan, 1992):

$$p\left(y = k \mid \boldsymbol{x}\right) = \frac{\exp(\boldsymbol{w}_k^\top \boldsymbol{x} + b_k)}{\sum_{j=1}^{C} \exp(\boldsymbol{w}_j^\top \boldsymbol{x} + b_j)},$$

with row weights $\boldsymbol{w}_k = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k$ and biases $b_k = \ln \pi_k - \frac{1}{2}\boldsymbol{w}_k^\top \boldsymbol{\mu}_k$, where π_k is the class prior. The empirical moment $(\hat{\pi}_k, \hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}})$ estimated from $\mathcal{D}$ yield the Maximum Likelihood Estimator which exactly has the form of a linear layer $\left(\boldsymbol{W}_{\mathrm{LDA}} = (\boldsymbol{w}_1, \ldots, \boldsymbol{w}_C)^\top, \boldsymbol{b}_{\mathrm{LDA}} = (b_1, \ldots, b_C)^\top\right)$ followed by the softmax operator. Indeed, as $n \to \infty$, this MLE coincides with the global minimizer of the cross-entropy loss over any linear model, making LDA the Bayes-optimal classifier in this setting (McLachlan, 1992; Papyan et al., 2020). This explains the success of last-layer linear probing on strongly pre-trained features (Kumar et al., 2022; Alain and Bengio, 2016): when frozen features are approximately class-conditional Gaussian distributed, fitting a linear head by cross-entropy is equivalent to computing the MLE in closed form.

Heteroscedastic Gaussian data. When the homoscedasticity assumption is relaxed, each class k has its own covariance $\boldsymbol{\Sigma}_k$:

$$\boldsymbol{x} \mid y = k \;\sim\; \mathcal{N}(\boldsymbol{\mu}_k,\, \boldsymbol{\Sigma}_k). \tag{35}$$

This is Quadratic Discriminant Analysis (QDA). Define the quadratic discriminant as

$$\mathrm{Q}_k(\boldsymbol{x}) = -\tfrac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_k) - \tfrac{1}{2}\ln|\boldsymbol{\Sigma}_k| + \ln \pi_k, \tag{36}$$

so that the posterior probability $p\left(y = k \mid \boldsymbol{x}\right) = \mathrm{softmax}\left(\mathrm{Q}(\boldsymbol{x})\right)_k$ (McLachlan, 1992). Under the heteroscedastic Gaussian assumption and in the limit $n \to \infty$, minimizing the cross-entropy loss drives the neural network's predictive distribution to this softmax (McLachlan, 1992).

Even when the Gaussian assumption is only approximate, LDA and QDA remain powerful and computationally efficient proxies as confirmed by the strong linear-probing results of recent foundation models (Oquab et al., 2024; Kumar et al., 2022). Figure 7 illustrates this equivalence on a toy example.

Cost of Homoscedastic Assumption. This subsection exhibits the cost to model the retain proxy by LDA rather than QDA.

Theorem 1 (Pythagorean Theorem of Information Csiszár (1975).)**.** *Let $\mathbb{P}$ be a closed convex set of probability distributions, $p_0 \notin \mathbb{P}$, and $p^\star = \arg\min_{p \in \mathbb{P}} D_{\mathrm{KL}}(p \,\|\, p_0)$ the I-projection of p_0 onto $\mathbb{P}$. Then for any $p \in \mathbb{P}$:*

$$D_{\mathrm{KL}}(p \,\|\, p_0) \geq D_{\mathrm{KL}}(p \,\|\, p^\star) + D_{\mathrm{KL}}(p^\star \,\|\, p_0)\,, \tag{37}$$

with equality when $\mathbb{P}$ is a linear family.

We apply this theorem class by class conditioning on $y = k$ for $k \in \{1, \ldots, C\}$. Let us define $\mathbb{P}_k$ as the family of probabilities with mean $\boldsymbol{\mu}_k$ and covariance $\boldsymbol{\Sigma}_k$:

$$\mathbb{P}_k \doteq \left\{P :\, \mathbb{E}_P[\boldsymbol{x}] = \boldsymbol{\mu}_k, \mathbb{E}_P[(\boldsymbol{x} - \boldsymbol{\mu}_k)(\boldsymbol{x} - \boldsymbol{\mu}_k)^T] = \boldsymbol{\Sigma}_k\right\}. \tag{38}$$

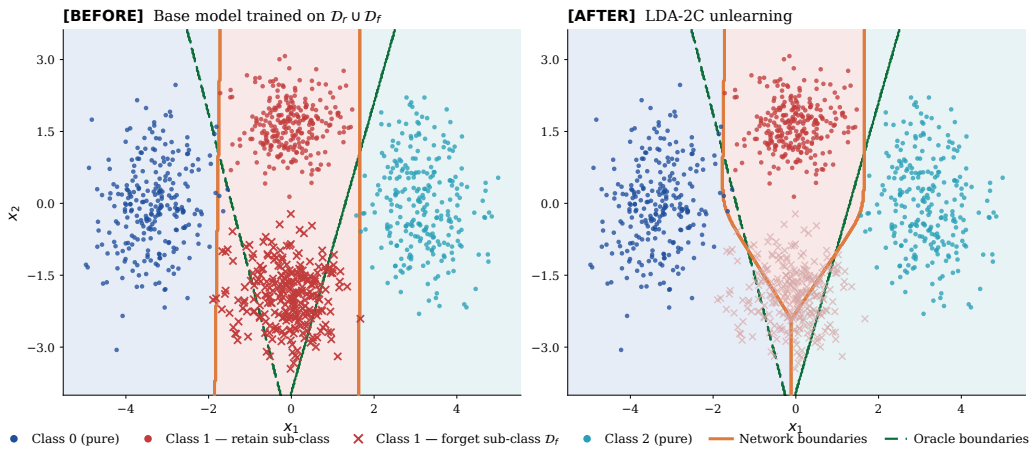


Figure 8: Illustration of the sub-class scenario over synthetic heteroscedastic Gaussian data. Class 1 is composed of two sub-classes, green and red points. The latter have to be unlearned.

This is a closed convex set of probabilities and a linear family in the sense of (37), as it is defined by linear equality constraints on the sufficient statistics ($P \mapsto \mathbb{E}_P[\boldsymbol{x}]$ and $P \mapsto \mathbb{E}_P[\boldsymbol{x}\boldsymbol{x}^\top]$ are linear functionals of P), so the Pythagorean equality holds exactly.

The I-projection of Amari (1983) of $p_0 = \hat{p}_r^{\mathrm{LDA}}(\boldsymbol{x} \mid y = k) \notin \mathbb{P}_k$ onto $\mathbb{P}_k$ is indeed $\hat{p}_r^{\mathrm{QDA}}(\boldsymbol{x} \mid y = k)$, *i.e.* $\mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$. We write $D_{\mathrm{KL}}(p \,\|\, p_0) = \mathbb{E}_p[-\log p_0(\boldsymbol{x})] - H(p)$. Since p_0 is a Gaussian distribution, $-\log p_0(\boldsymbol{x})$ is quadratic in $\boldsymbol{x}$ whose expectation depends only on the statistics $(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$. In other words, this cross-entropy term is constant over $\mathbb{P}_k$. Minimizing $D_{\mathrm{KL}}(p \,\|\, p_0)$ then amounts to maximzing the entropy $H(p)$ over $\mathbb{P}_k$, which is achieved by $\mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ Cover and Thomas (2006).

$$p^\star = \operatorname*{arg\,min}_{p \in \mathbb{P}_k} D_{\mathrm{KL}}\big(p \,\big\|\, \hat{p}_r^{\mathrm{LDA}}(\boldsymbol{x} \mid y = k)\big) = \hat{p}_r^{\mathrm{QDA}}(\boldsymbol{x} \mid y = k)\,.$$

Assuming that $p_r(\boldsymbol{x} \mid y = k) \in \mathbb{P}_k$, the equality holds exactly:

$$D_{\mathrm{KL}}^{\boldsymbol{x}|y=k}\big(p_r \,\big\|\, \hat{p}^{\mathrm{LDA}}\big) = D_{\mathrm{KL}}^{\boldsymbol{x}|y=k}\big(p_r \,\big\|\, \hat{p}^{\mathrm{QDA}}\big) + D_{\mathrm{KL}}^{\boldsymbol{x}|y=k}\big(\hat{p}^{\mathrm{QDA}} \,\big\|\, \hat{p}^{\mathrm{LDA}}\big)\,.$$

Lemma 4. *The KL between two Gaussians with the same mean $\boldsymbol{\mu}_k$ and different covariances $\boldsymbol{\Sigma}_k$ and $\boldsymbol{\Sigma}$ respectively:*

$$D_{\mathrm{KL}}(\mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \,\|\, \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma})) = \frac{1}{2}\left(\log \frac{|\boldsymbol{\Sigma}|}{|\boldsymbol{\Sigma}_k|} + \mathrm{Tr}(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_k) - d\right).$$

Finally, we link the above lemma to Prop. 3 with the following result:

Proposition 4. *The explicit cost of the homoscedastic assumption, i.e.* LDA *proxy, on the bound of Prop. 3.*

$$\mathbb{E}_{y\sim p_r} D_{\mathrm{KL}}^{\boldsymbol{x}|y}\big(p_r \,\big\|\, \hat{p}_r^{\mathrm{LDA}}\big) = \mathbb{E}_{y\sim p_r} D_{\mathrm{KL}}^{\boldsymbol{x}|y}\big(p_r \,\big\|\, \hat{p}_r^{\mathrm{QDA}}\big) + \frac{1}{2}\sum_{k=1}^{C} p_r(k)\left(\log \frac{|\boldsymbol{\Sigma}|}{|\boldsymbol{\Sigma}_k|} + \mathrm{Tr}(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_k) - d\right).$$

An unlearner knows exactly the cost to pay in term of divergence when reducing the computational cost of various inverse covariances.

F Exact Unlearning Signal with Empirical Distribution

What happens if we model the data structure with indicator function over the labeled data $(\boldsymbol{x}, y(\boldsymbol{x}))$?

We derive a closed-form solution and discuss its consequences for the unlearned model and for the distillation (4). We use the framework of mixture models described in Sect. 3; in particular the mixture $\hat{p}(\boldsymbol{x} \mid y) = \pi_r(y)\hat{p}_r(\boldsymbol{x} \mid y) + \pi_f(y)\hat{p}_f(\boldsymbol{x} \mid y)$ and posteriors (8).

The unlearner chooses the empirical measures for both proxies:

$$\hat{p}_r(\boldsymbol{x} \mid y) = \frac{1}{|\mathcal{D}_r(y)|}\mathbf{1}\{(\boldsymbol{x}, y) \in \mathcal{D}_r\},$$

and likewise for $\hat{p}_f$ on $\mathcal{D}_f$. Proxy $\hat{p}$ becomes a mixture of empirical measures on $\mathcal{D}$:

$$\hat{p}\,(\boldsymbol{x} \mid y) = \frac{\pi_r(y)}{|\mathcal{D}_r(y)|}\mathbf{1}\{(\boldsymbol{x}, y) \in \mathcal{D}_r\} + \frac{\pi_f(y)}{|\mathcal{D}_f(y)|}\mathbf{1}\{(\boldsymbol{x}, y) \in \mathcal{D}_f\}.$$

Since $\frac{\pi_r(y)}{|\mathcal{D}_r(y)|} = \frac{|\mathcal{D}_r(y)|}{|\mathcal{D}(y)||\mathcal{D}_r(y)|} = \frac{1}{|\mathcal{D}(y)|}$, we simply recover:

$$\hat{p}\,(\boldsymbol{x} \mid y) = \frac{1}{|\mathcal{D}(y)|}\mathbf{1}\{(\boldsymbol{x}, y) \in \mathcal{D}\},$$

which is the initial empirical measure. This is the same as doing *no* mixture.

For any $(\boldsymbol{x}, y) \in \mathcal{D}_r$, we have $\hat{p}_r(y) = p_r(y)$ and $\hat{p}_r(\boldsymbol{x}) = \sum_c p_r(c)\hat{p}_r(\boldsymbol{x} \mid c) = p_r(y)\hat{p}_r(\boldsymbol{x} \mid y)$. The Bayes inversion (8) leaves $\hat{p}_r(y \mid \boldsymbol{x}) = \hat{p}_r(\boldsymbol{x} \mid y)\, p_r(y)/\hat{p}_r(\boldsymbol{x}) = 1$. This also holds for $\hat{p}$.

$$\hat{p}_r(y \mid \boldsymbol{x}) = \mathbf{1}\{(\boldsymbol{x}, y) \in \mathcal{D}_r\}, \quad \hat{p}\,(y \mid \boldsymbol{x}) = \mathbf{1}\{(\boldsymbol{x}, y) \in \mathcal{D}\}.$$

This yields the following unlearning signal:

$$(\log \hat{p}_r - \log \hat{p})(\boldsymbol{x})_y = \log \frac{\mathbf{1}\big\{(\boldsymbol{x}, y) \in \mathcal{D}_r\big\}}{\mathbf{1}\big\{(\boldsymbol{x}, y) \in \mathcal{D}_r\big\} + \mathbf{1}\big\{(\boldsymbol{x}, y) \in \mathcal{D}_f\big\}}.$$

Since $\mathcal{D}_r \cap \mathcal{D}_f = \emptyset$, we obtain the closed form

$$(\log \hat{p}_r - \log \hat{p})(\boldsymbol{x})_y = \log \mathbf{1}\big\{(\boldsymbol{x}, y) \in \mathcal{D}_r\big\}. \tag{39}$$

The unlearn model $\log q$ completely annihilates the $(\boldsymbol{x}, y)$ data (logit set to $-\infty$) unless it belongs to the retain set where the logit is left unchanged.

No more dependency on η. For any $\eta > 0$, the $-\infty$ entries dominates the softmax of $\log q_\eta$ uniformly, leaving the truncated probit in the limit:

$$\forall \boldsymbol{x} \notin \mathcal{D}_r,\ q_\eta(\boldsymbol{x})_c = \begin{cases} 0 & c = y(\boldsymbol{x}), \\ f_{\boldsymbol{\theta}}(\boldsymbol{x})_c \,/\, (1 - f_{\boldsymbol{\theta}}(\boldsymbol{x})_{y(\boldsymbol{x})}) & c \neq y(\boldsymbol{x}). \end{cases} \tag{40}$$

The truncated probit (40) carries zero mass at y on the forget set. On the other hand, for all $\boldsymbol{x} \in \mathcal{D}_r$, $q_\eta(\boldsymbol{x}) = f_{\boldsymbol{\theta}}(\boldsymbol{x})$. The unlearned probit is η-independent.

Consequence on the distillation pipeline As it is, this correction makes little sense in practice, *i.e.* for unseen testing data. Yet, the KL distillation (4) over $\mathcal{D}$ remains well-posed because $D_{\mathrm{KL}}(q \,\|\, \mathrm{softmax}(\boldsymbol{\ell}_{\boldsymbol{\theta}+\delta\boldsymbol{\theta}}))$ uses q as the left argument and $0 \cdot \log 0 = 0$. The KL distillation drives the unlearned NN to get closer to the truncated probit (40), but *not* exactly equal to. This makes a strong decrease of the probit related to $y(\boldsymbol{x})$ for the forget data.

Empirical exact unlearning in $2C$ dimensions The previous truncation (40) is too aggressive, especially in the *Subclass* scenario. We lift this truncation from labels $y \in [\![C]\!]$ to 2C-dimensions labels $(y, s) \in [\![C]\!] \times \{r, f\}$. This amounts to canceling the mass probability of $(y(\boldsymbol{x}), f)$ for $\boldsymbol{x} \in \mathcal{D}_f$ and redistributing this quantity to the remaining cells (y, s). Each s-row needs a renormalization constant, equal to the probability $\pi_s(\boldsymbol{x})$ of a given sample $\boldsymbol{x}$ to be in the retain ($s = r$) or the forget set ($s = f$). The simplest inference uses a feature agnostic assumption: $\hat{p}\,(s \mid \boldsymbol{x}, y) \simeq \hat{p}\,(s \mid y) = \pi_s(y) = {}^{|\mathcal{D}_s(y)|}/_{|\mathcal{D}(y)|}$:

$$\pi_s(\boldsymbol{x}) = \sum_y \hat{p}\,(s \mid \boldsymbol{x}, y)\, \hat{p}\,(y \mid \boldsymbol{x}) \simeq \hat{\pi}_s(\boldsymbol{x}) \doteq \sum_y \hat{p}\,(y \mid \boldsymbol{x})\, \pi_s(y).$$

Since $\hat{p}\,(y \mid \boldsymbol{x}) = \mathbf{1}\big\{(\boldsymbol{x}, y) \in \mathcal{D}\big\} = \mathbf{1}\{y = y(\boldsymbol{x})\}$, then $\hat{\pi}_s(\boldsymbol{x}) = \pi_s(y(\boldsymbol{x}))$. One can retrieve the dimension C classifier:

$$q(\boldsymbol{x}) \doteq \hat{\pi}_r(\boldsymbol{x})\ f_{\boldsymbol{\theta}}(\boldsymbol{x}) + \hat{\pi}_f(\boldsymbol{x})\ q_f(\boldsymbol{x}),$$

with $q_f(\boldsymbol{x})$ defined as the previous the C dimension unlearned model (40). In the end, we have a feature-agnostic unlearned model using only the empirical distributions. Again, it makes little sense in practice, *i.e.* for unseen testing data. The KL distillation drives the unlearned NN to get closer to this target. This results in a slight decrease of the probit related to $y(\boldsymbol{x})$ for the forget data. This approximation makes experimentally a strong behavioral baseline for the subclass scenario.

G Our framework encompasses ULD, UCD, and T3-Unlearning

This appendix justifies the claim of Sect. 2 that our framework (3) generalises the logit-difference methods recently introduced for unlearning in LLM's setup. The transposition of our framework to LLMs gives $\boldsymbol{x}$ as the context, y as the next token, and C as the vocabulary size. For each paper, we identify our framework with the paper's notation. Of note, two logits differing by a constant vector produce the same probabilities of next token using the softmax operator.

ULD (Ji et al., 2024). It trains an expert assistant on the forget set. For the retain side it is equivalent to output the agnostic uniform law $\boldsymbol{u} \doteq \frac{1}{C}\mathbf{1}$. Its logit output is denoted ℓ_a and serves as a correction of the the initial model logit.

$$\ell_f = \boldsymbol{\ell}_{\boldsymbol{\theta}} - \alpha\, \ell_a \tag{41}$$

Adding the constant vector $\alpha \log \boldsymbol{u}$, we identify $\hat{p} = \mathrm{softmax}(\ell_a)$, $\hat{p}_r = \boldsymbol{u}$, and $\eta = \alpha$.

UCD (Suriyakumar et al., 2025). It uses a corrupted model A_{corr} on the forget set and into a safe model A_{clean} on the retain set.

$$\log P_{\mathrm{aligned}} \propto \log P_{\mathrm{corr}} - \alpha(\log A_{\mathrm{corr}} - \log A_{\mathrm{clean}}) \tag{42}$$

We identify $\hat{p} = A_{\mathrm{corr}}$, $\hat{p}_r = A_{\mathrm{clean}}$, $\eta = \alpha$. Our Prop. 2 is that same statement, but generalized to imperfect proxies.

T3 (Block et al., 2026). Its transformation use a temperature $T \geq 1$

$$\hat{p}_r^{(T)} \propto p^{1/T} \hat{f}, \tag{43}$$

and the classifier $\hat{f}$ estimates implicitely our unlearning signal. This is our framework applied twice:

(i) *Tempering*. We identify $\hat{p} = f_{\boldsymbol{\theta}}$, $\hat{p}_r = \boldsymbol{u}$, and $1 - \eta = 1/T$, since

$$\log f_{\boldsymbol{\theta}} + \eta(\log \boldsymbol{u} - \log f_{\boldsymbol{\theta}}) = \frac{1}{T} \log f_{\boldsymbol{\theta}} + \gamma \mathbf{1}$$

(ii) *Tilting*. Proxies are implicitly encoded into $\hat{f}(\boldsymbol{x}, y) \simeq p\,(r \mid \boldsymbol{x}, y) = \frac{p_r(y|\boldsymbol{x})p(r|\boldsymbol{x})}{p(y|\boldsymbol{x})}$ so that

$$\log \hat{f}(\boldsymbol{x}, y) = (\log \hat{p}_r\,(y \mid \boldsymbol{x}) - \log \hat{p}\,(y \mid \boldsymbol{x})) + \log \hat{p}\,(r \mid \boldsymbol{x})$$

with the implicit pair $(\hat{p}_r\,(y \mid \boldsymbol{x})\,, \hat{p}\,(y \mid \boldsymbol{x})) = (f_{\boldsymbol{\theta}}\,(y \mid \boldsymbol{x})\,\hat{f}(\boldsymbol{x}, y)/\hat{p}\,(r \mid \boldsymbol{x})\,,\ f_{\boldsymbol{\theta}}\,(y \mid \boldsymbol{x}))$ used with $\eta = 1$ and where

$$\hat{p}\,(r \mid \boldsymbol{x}) \doteq \sum_{c=1}^{C} f_{\boldsymbol{\theta}}\,(c \mid \boldsymbol{x})\,\hat{f}(\boldsymbol{x}, c). \tag{44}$$

Importantly, our procedure for $\eta_{\max}$ Alg. 2 needs to compute the log ratio, renormalised as a log shift of two distributions

$$\log \frac{\hat{p}_r\,(y \mid \boldsymbol{x})}{\hat{p}\,(y \mid \boldsymbol{x})} = \log \hat{f}(\boldsymbol{x}, y) - \log \sum_{c=1}^{C} f_{\boldsymbol{\theta}}\,(c \mid \boldsymbol{x})\,\hat{f}(\boldsymbol{x}, c) \tag{45}$$

and not both logs separately, so that it readily can be applied on the tilting of T3 Block et al. (2026).

Their choice of $\eta = 1$ is justified only with ideal estimation of $\log \hat{f}$, and we infer that it is why they needed to introduce the additional tempering, instead of deriving an $\eta_{\max}$ potentially greater than 1. Then, putting $\eta = \eta_{\max}$ instead of $\eta = 1$ ensures the behavior in case of approximated tilting $\hat{f}$.

Conclusion. Our framework encompasses these recent works. Yet they use hyper parameters $(\alpha, T, \eta = 1)$ fixed heuristically or by grid search, whereas we provide a theoretically grounded procedure (see Alg. 2).

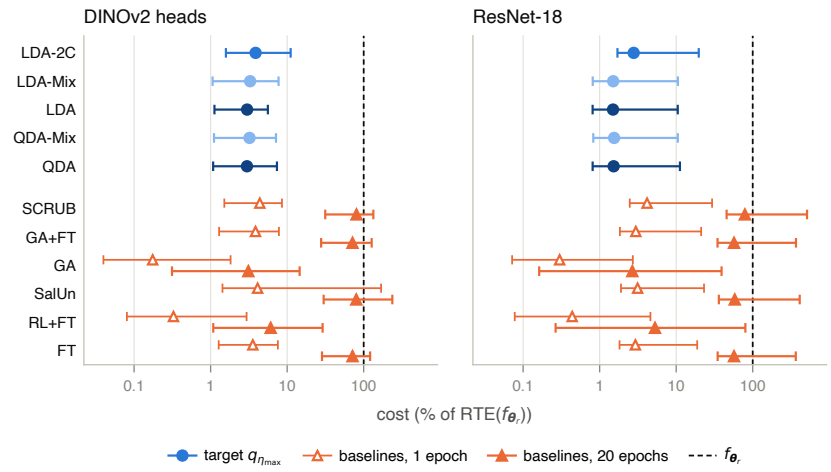


Figure 9: Comparison of the runtime of deriving our black-box unlearned model to the runtime of the baselines. For ResNet-18 : $k = 512$ on layer $\ell = 3$

H HANDLING LARGE NN

This appendix shows that scaling our framework to networks whose hidden representations are too high-dimensional for a direct LDA/QDA fit requires no theoretical change. It is only a matter of choice of the proxies.

Composed proxy The whole machinery of Sect. 3 only requires modeling the input distribution so that the proxy shift can be computed in closed form. Up to now we instantiated this proxy directly on $\boldsymbol{x}$ via LDA or QDA for instance. When $\boldsymbol{x}$ lives in a space too large for a class-conditional Gaussian fit, we simply compose:

$$\hat{p}\,(\boldsymbol{x} \mid y) \;\doteq\; \hat{p}_0(F(\boldsymbol{x}) \mid y)\,, \qquad F\colon \mathbb{X} \to \mathbb{R}^k, \tag{46}$$

where $\hat{p}_0$ is the standard LDA or QDA proxy on the space $F(\mathbb{X})$ and F is any fixed injection from the high-dimensional input space onto a low-dimensional target. The retain proxy $\hat{p}_r$ is built identically with the same F. Plugging (46) into the framework yields a logit shift that is computed in $\mathbb{R}^k$.

The assumptions and the bounds of Sect. 3 carry over verbatim. Any F that yields a low-dimensional representation on which LDA/QDA is accurate is admissible. Indeed, our framework never constrains the form of the proxy: it only asks for a pair $(\hat{p}, \hat{p}_r)$ of models of the class-conditional law whose posterior log-ratio is available in closed form. For a fixed, data-independent F, the composition $\hat{p}\,(\boldsymbol{x} \mid y) \doteq \hat{p}_0(F(\boldsymbol{x}) \mid y)$ is such a model, hence a perfectly legitimate proxy.

Advantages. The practical consequence matters more than the formal one. A sketch F that destroys the inter-*Class* geometry or a projection that loses too much information surfaces before any update is applied: Alg. 2 returns $\eta_{\max} = 0$ and the network is left untouched. In other words, a bad projection is detected, not silently suffered.

Projections help us compete with the iterative methods not only in terms of performance, but also in runtime ans shown in Fig. 9.

Practical instance for ResNet-18. The specific F we use in the ResNet-18 experiments (Sect. 5) is anecdotal, it is one convenient choice among many. We take

$$F(\boldsymbol{x}) \;=\; \boldsymbol{\Omega}\,\phi^{(3)}(\boldsymbol{x}), \tag{47}$$

where $\phi^{(3)}$ is the activation just after the third residual stage ($\dim \sim 8{\times}10^3$ at our resolution) and $\boldsymbol{\Omega} \in \mathbb{R}^{k \times d}$ ($k = 512$) is a deterministic semi-orthogonal Johnson–Lindenstrauss sketch (Johnson and Lindenstrauss, 1984; Dasgupta and Gupta, 2003) obtained as the orthogonal factor of a Gaussian matrix via QR decomposition, with a fixed seed shared across all proxy variants.

Where to tap the proxies. The hook depth ($\phi^{(3)}$) is the standard mid-level/pre-classification choice: earlier stages have too small a receptive field, producing very long features which are far from a Gaussian distribution. Later ones already encode logits and would make the proxies redundant with $\ell_{\boldsymbol{\theta}}$ and lack the information of membership. Fig. 10 show that the resulting target is very dependant on the layer quality. Near the output layer, the membership information is more compress and the unlearning signal is indistinguishable. Another phenomenon is that the best layer depends on the task, and its granularity. Figure 10 highlight that the behavior is very different between the *Class* and *Subclass* scenarios. Along the network, the membership information is compressed until it vanishes. *Subclass* scenario needs earlier layers to perform well because the information to extract is thinner: *Subclass* of a super-class.

Section 5 only uses the layer $\ell = 3$ and projection dimension $k = 512$. This choice is deliberately shown to be sub-optimal in the *Subclass* scenario (see. Fig. 10). The framework would benefit from reducing the dimension k and selecting an earlier layer. This illustrates how the problematic of unlearning is shifting towards features representation/selection. In the worst case, our procedure is safe as Alg. 2 returns $\eta_{\max} = 0$ which leave the initial model unchanged. The user may then test another projection.

Choice of features dimensions. Intuitively, one has to match the feature dimensions with the quantity of data available for identifying the model.

For Gaussian proxies, a $k \times k$ class covariance needs at least k samples to be non-singular so the per-*Class* data budget $|\mathcal{D}(y)|$ caps k. On the CIFAR-10 *Subclass* task, whose two superclasses hold $25,000$ images each, $k = 512$ is comfortable. On CIFAR-100, there are 500 images per class, *i.e.* less than k. A proxy sharing the covariance matrices is preferred. Indeed, the limiting factor is the size of the forget set, whose per-*Class* counts $|\mathcal{D}_f(y)|$ might be very small. This is one more reason to prefer the single shared covariance of `LDA-2C` or `LDA` to the per-state or per-*Class* covariances of `LDA-Mix`/`QDA-Mix`.

Conclusion. This framework scales to arbitrary backbones for a fixed cost in $O(k)$ because the proxy fit is performed in $\mathbb{R}^k$ regardless of the input dimension. The features is what matters most in practice when modeling the data in order to create proxies. Operator F is the only choice that needs revisiting for a new architecture: any low-dimensional representation on which LDA/QDA is reasonable. The tap layer is more important than the dimension of the projection (see Fig. 10).

I PRIVACY AUDITING

Our unlearning procedure is designed to bring guarantees on the behavior of the unlearned model. It is not a priori privacy oriented. Yet, the following section discusses the connection between privacy auditing and behavioral guarantees. The other sections experimentally show that our unlearning procedure is beneficial from the privacy point of view.

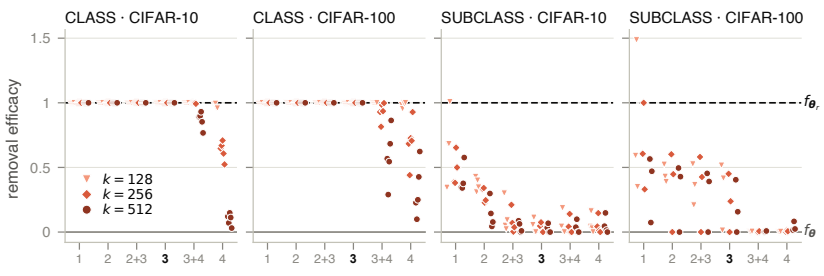


Figure 10: Removal efficacy, defined as $\big(\mathrm{Acc}^{\mathrm{f}}(f_{\boldsymbol{\theta}}) - \mathrm{Acc}^{\mathrm{f}}(q)\big)/\big(\mathrm{Acc}^{\mathrm{f}}(f_{\boldsymbol{\theta}}) - \mathrm{Acc}^{\mathrm{f}}(f_{\boldsymbol{\theta}_r})\big)$, vs. layer ℓ (or combination) for the proxy `LDA-2C`. Projection dimension $k \in \{128, 256, 512\}$. **Bold**: Sect. 5 uses layer $\ell = 3$.

I.1 Membership Inference Attacks and KL divergence

This section relates the KL divergence, the core metric in this paper, to privacy auditing.

Imagine that an attacker has access to a classifier in a black box. He knows that this black box contains either the ideal model retrained from scratch $f_{\boldsymbol{\theta}_r}$ or the unlearned NN $f_{\boldsymbol{\theta}_u}$. He queries the black box with some random data $\boldsymbol{x}$ and observes the output $\hat{y}$, the predicted label. Of note, this is a worst-case scenario for the attacker. MIAs in the literature agree on a more favorable scenario where the attack has access to the probit or logit vector.

This section relates D_{KL}^f (or D_{KL}^t), as reported in the experimental results, to the number of queries necessary to identify the classifier in the black-box with a given reliability.

For a given input $\boldsymbol{x}$, we have the following hypothesis:

$$\mathcal{H}_0 : \hat{y} \sim f_{\boldsymbol{\theta}_r}(y \mid \boldsymbol{x}) \quad \text{versus} \quad \mathcal{H}_1 : \hat{y} \sim f_{\boldsymbol{\theta}_u}(y \mid \boldsymbol{x})$$

Let us denote the probability of a false positive by α and the probability of a false negative by β. The attacker's test outputs a binary decision $\mathrm{d} \in \{0, 1\}$, which has the following Bernoulli distribution:

$\mathcal{H}_0$**:** The test makes a *false positive* whenever $\mathrm{d} = 1$. Then, $\mathrm{d} \sim \mathcal{B}(\alpha)$.

$\mathcal{H}_1$**:** The test makes a *false negative* whenever $\mathrm{d} = 0$. Then, $\mathrm{d} \sim \mathcal{B}(1-\beta)$.

Suppose that the attacker designs the test to achieve an Equal Error Rate, *i.e.* $\alpha = \beta$, since the error rates α and β matter equally.

The data processing theorem states that any processing of the data $\hat{y}$ reduces the KL divergence. The attacker's test is a processing; therefore:

$$\begin{aligned} D_{\mathrm{KL}}^{y|\boldsymbol{x}}(p_r \,\|\, f_{\boldsymbol{\theta}_u}) &\geq D_{\mathrm{KL}}(\mathcal{B}(\alpha) \,\|\, \mathcal{B}(1-\beta)) = (1-\alpha)\log\left(\frac{1-\alpha}{\beta}\right) + \alpha\log\left(\frac{\alpha}{1-\beta}\right) \\ &\overset{(\alpha=\beta)}{=} (1-2\alpha)\log\left(\frac{1-\alpha}{\alpha}\right). \end{aligned}$$

This inequality is at the root of the Stein lemma.

Querying $N > 1$ inputs brings more information. Provided they are independent, their KL divergences sum up. This sum can be written as $N \cdot \mathbb{E} D_{\mathrm{KL}}^{y|\boldsymbol{x}}(f_{\boldsymbol{\theta}_r} \,\|\, f_{\boldsymbol{\theta}_u})$, where the expectation is indeed an average over the queries as reported experimentally in Sect. 5. At the end, we get a necessary number of queries to reach a reliability given by $1-\alpha$:

$$N \geq \frac{(1-2\alpha)\log\left({}^{1-\alpha}/_{\alpha}\right)}{\mathbb{E} D_{\mathrm{KL}}^{y|\boldsymbol{x}}(f_{\boldsymbol{\theta}_r} \,\|\, f_{\boldsymbol{\theta}_u})}.$$

This number comes with two flavors. The attacker may query test data (resp. forget data) and we plug D_{KL}^t (resp. D_{KL}^f) in the formula. For instance, for $\alpha = 1/1\,000$, $N_t = 138$ for $D_{\mathrm{KL}}^t = 0.05$, $N_f = 69$ for $D_{\mathrm{KL}}^f = 0.1$. The key question is whether $|\mathcal{D}_f| < N_f$; if so, the attacker cannot distinguish the two classifiers with forget data, even if he knows all of them.

Conclusion. The basic rationale of this study is the following: If an attacker is not able to distinguish $f_{\boldsymbol{\theta}_u}$ from a model $f_{\boldsymbol{\theta}_r}$, which never processed $\mathcal{D}_f$, then he can not infer anything about the forget set. This rationale is not true because this analysis is not a privacy audit. It delivers no per-example statement and no (ε, δ) statement either.

First, the quantity N_f is based on an *average* over the forget set. This is a so-called *population* MIA in (Hayes et al., 2025) or *average-case* evaluation in (Naderloui et al., 2025). A handful of forget samples may remain perfectly identifiable, typically the ones per-example audits of the U-LiRA family are able to expose.

On the second point, our unlearning is indeed not compliant with Differential Privacy. If the attacker knows $\mathcal{D}$, $f_{\boldsymbol{\theta}}$, $f_{\boldsymbol{\theta}_u}$, and the unlearning procedure, he can theoretically identify $\mathcal{D}_f$ by an exhaustive search because our unlearning is mainly deterministic. Randomness happens only in the last optional distillation step.

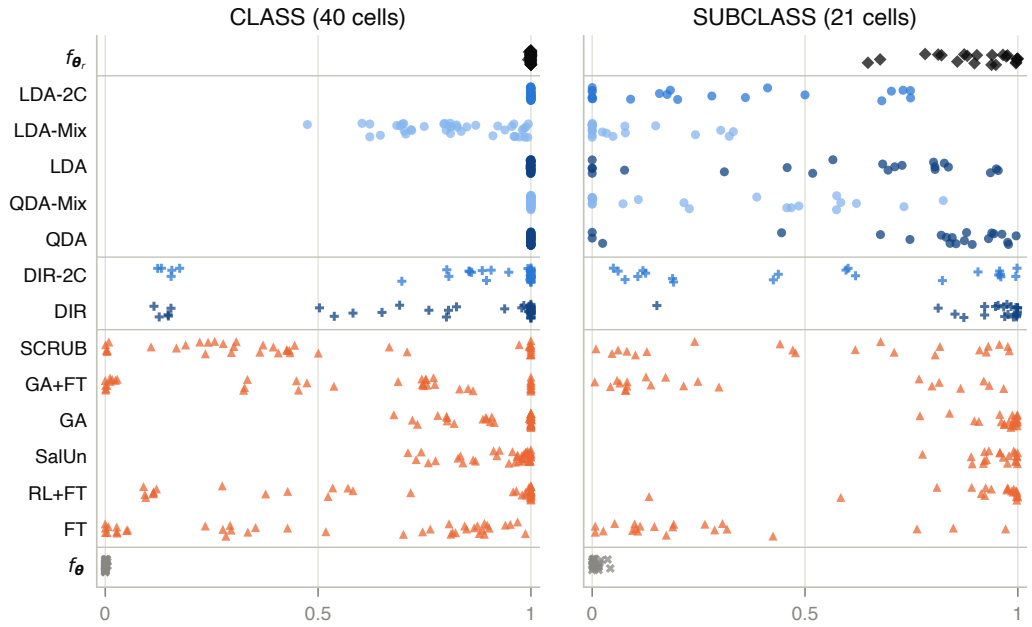


Figure 11: X-axis is the fraction of samples in $\mathcal{D}_f$ that are not flagged as training data. One mark per run across scenarios and architectures.

I.2 Per-sample audit

We evaluate a per-sample membership inference attack which decide that the sample $\boldsymbol{x} \in \mathcal{D}_f$ is not a training data if $D_{\mathrm{KL}}^{y|\boldsymbol{x}}(f_{\boldsymbol{\theta}_r} \,\|\, f_{\boldsymbol{\theta}_u}) < D_{\mathrm{KL}}^{y|\boldsymbol{x}}(f_{\boldsymbol{\theta}} \,\|\, f_{\boldsymbol{\theta}_u})$. It is our metric "Audit" in table Tab. 2. Audit measures the percentage of forget data that verifies this inequality. As observed in Sect. 5.3, Fig. 11 confirms that the *Subclass* scenario is more difficult. The difficulty transfers from reducing the KL on average, to reducing it per sample. It is also tangible when $f_{\boldsymbol{\theta}_u} = f_{\boldsymbol{\theta}_r}{}'$, a model trained from scratch independently of $f_{\boldsymbol{\theta}_r}$ (first line of Fig. 11). It is measuring something different than our Prop. 1 or our other KL metrics. For instance GA performs well overall on this audit while destroying the model.

On this audit, the naive proxies seem to be stronger than the 2C construction. Our explanation is as follows: the identification of $\hat{p}_r$ from naive constructions never sees data from the forget set. On the contrary, the 2C construction by design create a single fit where the identification of $\hat{p}_r$ sees forget data. The audit suggests that, for privacy purposes, the $\hat{p}_r$ may not be contaminated with forget data.

Scope of this audit. This per-sample test is weaker than a U-LiRA audit Hayes et al. (2025) and should not be read as a privacy guarantee. As discussed in the previous section, our target is deterministic given the proxies, so an attacker knowing $\mathcal{D}$, $f_{\boldsymbol{\theta}}$ and the procedure could in principle identify $\mathcal{D}_f$.