Delete and Retain: Efficient Unlearning for Document Classification

Aadya Goel[1] Mayuri Sridhar[2]

[1]Acton-Boxborough Regional High School [2]MIT

Abstract

Machine unlearning aims to efficiently remove the influence of specific training data from a model without full retraining. While much progress has been made in unlearning for LLMs, document classification models remain relatively understudied. In this paper, we study class-level unlearning for document classifiers and present **Hessian Reassignment**, a two-step, model-agnostic solution. First, we perform a single influence-style update that subtracts the contribution of all training points from the target class by solving a Hessian–vector system with conjugate gradients, requiring only gradient and Hessian-vector products. Second, in contrast to common unlearning baselines that randomly reclassify deleted-class samples, we enforce a decision-space guarantee via Top-1 classification. On standard text benchmarks, **Hessian Reassignment** achieves retained-class accuracy close to full retrain-without-class while running orders of magnitude faster. Additionally, it consistently lowers membership-inference advantage on the removed class, measured with pooled multi-shadow attacks. These results demonstrate a practical, principled path to efficient class unlearning in document classification.

Keywords: Machine Unlearning · Document Classification · Hessian · Data Privacy · Forgetting

1 Introduction

Modern machine learning systems increasingly operate under legal and user-driven deletion requests, including *right-to-be-forgotten* [1] provisions in regulations such as CCPA [2] and GDPR [3]. Exact unlearning by retraining from scratch is conceptually simple but often impractical for production models due to compute budgets and continual data churn. This tension has motivated *machine unlearning* [4], algorithms that remove the influence of specified training data from a deployed model without full retraining [5, 6, 7]. While prior work has largely considered removing individual points or small subsets via influence approximations [8], gradient-based adjustments [7], or partitioned training [5], these designs do not directly exploit additional *structure* in the deletion request.

We study the setting where an entire class must be forgotten in a trained document classifier. This situation arises when a label is deemed sensitive, a cohort associated with a category requests erasure, or other related events. After unlearning, the model must still produce predictions over the remaining labels for all inputs, including documents that originally belonged to the deleted class.

While existing methods remove the influence of the target class, they ignore the score ordering learned by the original model, degrading model utility [9]. Instead, we propose unlearning through top-1 reclassification.

Our key observation is that a trained document classifier already encodes a coherent ranking over these non-target labels for each input. We design a two-step, model-agnostic procedure that preserves this structure while removing the influence of the target class. We apply a single second-order "downweight" correction step and then do deterministic reassignment.

This paper makes the following contributions:

- We formalize *class-level unlearning* for document classifiers and identify the decision-space ambiguity it introduces.
- It introduces **Hessian Reassignment**, a simple two-step unlearning procedure that combines a single class-level Hessian downweight update with a deterministic next-top-1 decision rule over non-target labels.
- It shows empirically, on standard text benchmarks, that this procedure matches complete retraining on retained-class accuracy while substantially improving privacy and utility over Random Relabeling..

2 Related Work

2.1 Machine Unlearning Frameworks

Machine unlearning methods aim to remove the influence of specific training data without retraining from scratch. Several lines of work study scalable exact procedures for deep models [10], feature- and label-level removal [11], and the utility–complexity trade-offs of unlearning under distribution shift [12]. Other work emphasizes practical challenges around data quality, storage, and access, treating unlearning as a systems and data-management problem as much as an algorithmic one [13].

In language models, unlearning has been used for privacy-preserving fine-tuning [14], probing what information remains after deletion [15], analyzing the impact of pretraining data via removal [16], and efficiently forgetting specific content or behaviors in large pretrained models [27, 28]. Overall, these frameworks primarily address point-level or subset-level deletion in deep architectures. In contrast, this paper considers removing an entire class from a document classifier while preserving the structure of predictions over the remaining labels.

2.2 Training Data Influence and Second-Order Methods

Training-data influence methods use information about the local *curvature* of the loss landscape, captured by the Hessian, to approximate how adding or removing examples would shift model parameters. One line of work estimates influence by tracing gradient descent trajectories [17], while another develops second-order group influence functions that analyze how sets of examples affect predictions via Hessian-based approximations [18]. Subsequent studies show that such estimates can be fragile in deep models and survey techniques for more robust influence analysis across architectures [19, 20].

Influence-style ideas have also been applied directly to unlearning, for example by tracing influential training data in language models [21] or using differential gradient approximations to scale influence estimation to larger systems [22]. The approach in this paper is conceptually related in that it uses second-order information, but rather than estimating pointwise influence or tracing training trajectories, it applies a single class-level Hessian downweight update that removes the aggregate contribution of all deleted-class examples in one step.

2.3 Unlearning in NLP and Document Models

Beyond generic unlearning frameworks, there is a growing body of work focused on natural language processing and large language models. Several methods use machine unlearning to remove memorized content, sensitive information, or undesired behaviors from pretrained language models [14, 16, 27, 28]. These approaches often operate at the level of model parameters, adapters, or task-specific heads for sequence models, and are typically designed for transformer-based architectures.

The setting studied in this paper is different: classical document classification with TF–IDF features and a linear-softmax head. While the techniques developed for language models could in principle be adapted to simpler pipelines, existing methods do not exploit the specific structure of class-level deletions in multi-class document classifiers. Here, the goal is to remove a full label and all of its training examples while preserving the model's learned ordering over the remaining labels. Unlike point deletions, class-level unlearning simultaneously alters the empirical label distribution and the geometry of multiple decision boundaries.

A central design choice here is how the model should behave on deleted-class documents. A simple and commonly referenced approach in document classification is to *randomly reassign* removed samples across the remaining classes ("Random Relabeling") [9]. Although easy to implement, this destroys the score ordering learned by the original model, often degrading utility and inducing unstable decision regions. Our proposed method addresses this by combining a deterministic next-top-1 decision rule with a single Hessian-based update.

2.4 Membership Inference Attack

Membership inference attacks (MIAs) evaluate whether a model reveals if a particular data point was part of its training dataset. A central framework is the *shadow-model* approach, in which an adversary trains auxiliary models to mimic a held-out target model and then uses confidence scores to distinguish members from non-members [23]. Follow-up work generalizes these ideas to a broader range of models and data regimes and shows that overfitting strongly correlates with privacy risk, and that membership inference remains a concern even for synthetic or heavily processed data [24, 25, 26].

These results motivate our privacy metrics. By measuring membership-inference ROC–AUC on both retained-class and deleted-class examples, we can quantify how much the proposed unlearning procedure reduces membership leakage, especially for the class that is supposed to be "forgotten."

3 Problem Formulation

Setting and notation. Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$ be a training corpus with documents $x_i \in \mathcal{X}$ and labels $y_i \in \mathcal{Y} = \{1, \ldots, K\}$. A classifier with parameters W produces class probabilities

$$p_W(y \mid x) = \operatorname{softmax}\big(s_W(x)\big)_y,$$

where $s_W(x) \in \mathbb{R}^K$ are model scores (architecture-agnostic). The empirical objective is

$$\ell(W; \mathcal{D}) = \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}\big(s_W(x_i), y_i\big) + \Omega(W),$$

with a standard loss $\mathcal{L}$ (e.g., cross-entropy) and regularizer Ω. Let the following denote the trained model

$$W^\star = \arg\min_W \ell(W; \mathcal{D}).$$

Class-level deletion. Given a target class $c \in \mathcal{Y}$, the request is to *forget* all its training examples $\mathcal{D}_c = \{(x_i, y_i) : y_i = c\}$.

The retained dataset and label set are $\mathcal{D}_{-c} = \mathcal{D} \setminus \mathcal{D}_c$ and $\mathcal{Y}_{-c} = \mathcal{Y} \setminus \{c\}$, respectively.

Unlearning objective. An unlearning operator $\mathcal{U}_c$ maps $(\mathcal{D}, W^\star)$ to unlearned model $\widetilde{W}$ such that:

1. **Deletion consistency.** On any evaluation set supported on $\mathcal{Y}_{-c}$, the performance of $\widetilde{W}$ matches that of the *complete retraining* solution

$$W^\star_{-c} = \arg\min_W \ell(W; \mathcal{D}_{-c}).$$

2. **Viable predictions for removed examples.** For all x originally labeled c, $\widetilde{W}$ produces a distribution over $\mathcal{Y}_{-c}$ (i.e., does not route probability mass to the removed class).

3. **Privacy after deletion.** Membership information about $\mathcal{D}_c$ is obscured under standard attacks (e.g., MIA), ideally approaching the attack performance against $W^{\star}_{-c}$.

We measure (1) via accuracy on $\mathcal{Y}_{-c}$, (2) via probability diagnostics over $\mathcal{Y}_{-c}$ (margins, KL), and (3) via removed-class MIA AUC.

Structure-Preserving Reassignment via Next Top-1 For each deleted-class document $x \in \mathcal{D}_c$, define the *next top-1* label under the *pre-unlearning* model $W^{\star}$:

$$\phi_{-c}(x) \;=\; \arg\max_{y \in \mathcal{Y}_{-c}} p_{W^{\star}}(y \mid x).$$

This answers the counterfactual: *If class c had never existed, which class would the deleted-class document be classified to?* We refer to $x \mapsto \phi_{-c}(x)$ as a *structure-preserving reassignment*, since it preserves the learned score ordering among non-removed classes.

Golden Standard We treat $W^{\star}_{-c}$ as the "Golden Standard" reference. The ideal unlearned model W_{-c} is defined as the classifier retrained from scratch on D_{-c}.

For a test distribution $\mathcal{T}$ over $\mathcal{X} \times \mathcal{Y}_{-c}$, a utility metric U (e.g., accuracy) and a privacy metric P (e.g., MIA AUC on $\mathcal{D}_c$), we desire

$$U(\widetilde{W}; \mathcal{T}) \approx U(W^{\star}_{-c}; \mathcal{T}) \quad \text{and} \quad P(\widetilde{W}) \approx P(W^{\star}_{-c}).$$

State-of-the-Art Baseline: Random Relabeling The current state-of-the-art class-level baseline in document classification unlearning is *random relabeling* [9], where each deleted-class document $x \in \mathcal{D}_c$ is reassigned a label sampled uniformly from the remaining set $\mathcal{Y}_{-c}$ and the model is fine-tuned or retrained on the modified dataset.

By construction, this scheme treats all non-c labels as equally plausible for deleted examples, ignoring the model's pre-existing ranking over $\mathcal{Y}_{-c}$ and replacing it with independent random targets. In other words, it injects heavy label noise concentrated on the forget set.

Prior work has noted that such relabeling-based unlearning can impair generalization or cause over-forgetting in deep models [9, 29, 30], and in our experiments (Section 6) we likewise observe that random relabeling often trades away retained-class utility to achieve privacy on the deleted class.

Requirements Specific to Class-Level Deletions

1. **Order preservation.** For $x \notin \mathcal{D}_c$, the ranking over $\mathcal{Y}_{-c}$ under $\widetilde{W}$ remains close to that under $W^{\star}$.

2. **Privacy focus.** Post-deletion attack advantage on $\mathcal{D}_c$ should approach chance (AUC ≈ 0.5), while retaining utility on $\mathcal{D}_{-c}$.

These properties define the target behavior independently of architecture and guide both our method and evaluation.

4 Methodology

Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$ be a labeled corpus with $y_i \in \mathcal{Y}$ and model parameters θ trained to minimize a regularized empirical loss

$$\mathcal{L}(\theta) = \sum_{i=1}^{n} \ell(x_i, y_i; \theta) + \Omega(\theta).$$

A deletion request specifies a target class $c \in \mathcal{Y}$. After unlearning, the model should (i) forget membership information tied to class c and (ii) preserve predictions among the remaining labels $\mathcal{Y}_{-c}$. We report utility as test accuracy excluding class c, and privacy via retained-class membership-inference AUC.

4.1 Top-1 classification reassignment

For each training point with $y_i = c$, define its next top-1 label under the pre-unlearning model:

$$r_i \;=\; \arg\max_{k \neq c} s_k(x_i; \theta),$$

where s_k are the model scores (e.g., logits). This deterministic rule preserves the model's score ordering among non-c classes and provides a stable target after deletion. We use this rule to define target reassignment behavior for deleted-class points.

4.2 Hessian downweight update

Recall the regularized empirical loss $\nabla\mathcal{L}(\theta)$. Deleting class c replaces $\mathcal{L}$ by

$$\mathcal{L}^{(-c)}(\theta) = \sum_{i \notin \mathcal{I}_c} \ell(x_i, y_i; \theta) + \Omega(\theta), \qquad \mathcal{I}_c\{\, i : \, y_i = c \,\}.$$

The new optimizer $\theta^\star$ satisfies

$$\nabla\mathcal{L}^{(-c)}(\theta^\star) = 0.$$

A second-order Taylor expansion of $\nabla\mathcal{L}^{(-c)}$ around θ gives

$$\nabla\mathcal{L}^{(-c)}(\theta^\star) \approx \nabla\mathcal{L}^{(-c)}(\theta) + H(\theta)\,(\theta^\star - \theta) = -\,g_c(\theta) + H(\theta)\,\Delta\theta,$$

where $H(\theta) := \nabla^2 \mathcal{L}(\theta)$[1] and

$$g_c(\theta) := \sum_{i \in \mathcal{I}_c} \nabla \ell(x_i, y_i; \theta)$$

is exactly the gradient contribution of the deleted class. Setting the linearized gradient to zero yields the *inexact Newton* step

$$\Delta\theta \;=\; H(\theta)^{-1} g_c(\theta), \qquad \theta' \;=\; \theta \;-\; \Delta\theta. \tag{1}$$

We call Eq. (1) a *Hessian downweight*. It subtracts (downweights) the first- and second-order influence of the deleted terms from the optimality conditions without performing complete retraining.

In Newton methods, $H(\theta)^{-1}$ maps a desired change in the gradient (*here*, removing g_c) to a parameter move that cancels it to first order. Intuitively, H^{-1} rescales each parameter direction by the local curvature: flat directions (small eigenvalues) require larger moves; stiff directions require smaller ones. Under a second-order optimization, θ' is the first order approximation to the optimizer obtained by retraining on $\mathcal{D}_{-c}$.

4.3 Instantiation for multinomial logistic regression

For a linear-softmax classifier with weights $W \in \mathbb{R}^{K \times d}$ (no intercept), let $P = \mathrm{softmax}(XW^\top)$. The gradient contribution of one sample (x_i, y_i) is

$$\nabla_W \ell_i \;=\; (p_i - e_{y_i})\, x_i^\top,$$

and the Hessian-vector product with $V \in \mathbb{R}^{K \times d}$ is

$$H[V] \;=\; \Big(\big(P \odot U\big) - \big(P \odot (s\, \mathbf{1}_K^\top)\big) \Big)^\top X \;+\; \lambda V,$$

where

$$U = XV^\top \in \mathbb{R}^{n \times K}$$

and $s \in \mathbb{R}^n$ has entries

$$s_i = \sum_{k=1}^{K} P_{ik}\, U_{ik}$$

(i.e., the per-row sum over classes). Here λ comes from the ℓ_2 regularizer. We compute $\Delta = H^{-1} G_c$ via conjugate gradients using only HVPs, and set $W' = W - \Delta$.

[1] Using $H(\theta)$ rather than $\nabla^2 \mathcal{L}^{(-c)}(\theta)$ yields the same first-order correction at θ and is more stable in practice.

4.4 Zeroing the removed output at release

At deployment we make the removed class inert by nulling its output channel (e.g., zeroing the corresponding logit row in a linear-softmax head) and renormalizing probabilities over $\mathcal{Y}_{-c}$. This prevents any downstream consumer from selecting class c.

5 Baselines & Evaluations

5.1 Baselines

Hessian Reassignment We will refer to our algorithm as Hessian Reassignment throughout the rest of this paper. This represents the Hessian downweight update (Sec. 4) with top-1 classification reassignment and a zeroed output for the removed class at release.

Random Relabeling (state-of-the-art baseline). All training points with $y_i = c$ are reassigned uniformly at random to labels in $\mathcal{Y}_{-c}$, after which the model is fine-tuned/updated on the modified corpus. This baseline ignores the original score ordering and serves as the method we aim to outperform. Note that random relabeling keeps all datapoints with $y_i = c$ and serves as an approximate unlearning mechanism rather than a deletion-based ground truth.

Complete retraining (Golden Standard). We remove all training examples with label c and retrain the classifier from scratch on D_{-c}. This produces the ideal counterfactual model W_{-c} which represents the exact behavior the system would exhibit had class c never been part of the training set.

5.2 Metrics

5.2.1 Model Utility

We report two accuracy-style utilities:

Accuracy excluding the removed class. Let $\widehat{y}(x)$ be the post-unlearning prediction over labels $\mathcal{Y}_{-c}$ (the removed output is made inert and probabilities re-normalized). We compute

$$\mathrm{Acc}_{\setminus c} = \frac{1}{|\{i : y_i \neq c\}|} \sum_{i:\, y_i \neq c} \mathbf{1}\{\widehat{y}(x_i) = y_i\}.$$

Agreement on the removed class w.r.t. the golden standard. Let $\mathcal{I}_c = \{i : y_i = c\}$ be the indices of the deleted class. Let $\widehat{y}^{\,\mathrm{ours}}(x)$ denote our model's prediction after unlearning and $\widehat{y}^{\,\mathrm{GS}}(x)$ denote the prediction from *complete retraining* model W_{-c} (Golden Standard). We measure how well our method reproduces the golden standard on the removed-class documents:

$$\mathrm{Agree}_c = \frac{1}{|\mathcal{I}_c|} \sum_{i \in \mathcal{I}_c} \mathbf{1}\big\{\widehat{y}^{\,\mathrm{ours}}(x_i) = \widehat{y}^{\,\mathrm{GS}}(x_i)\big\}\,.$$

5.2.2 Privacy (Membership Inference).

We evaluate membership–inference risk with a fixed attacker trained on shadow models. For a target model after removing class c, let $p(x) \in \Delta^{K-1}$ denote the prediction vector over the remaining classes (i.e., we drop class c and renormalize). The attacker uses a feature vector $\phi(x)$ that concatenates several statistics derived from the model's prediction on input x. Let $\|$ denote vector concatenation. Then

$$\phi(x) \;=\; \big[\, p(x) \,\|\, H\big(p(x)\big) \,\|\, -\log p_{\tilde{y}}(x) \,\|\, \mathrm{gap}\big(p(x)\big) \,\big],$$

where entropy is

$$H(p) = -\sum_j p_j \log p_j,$$

$\tilde{y}$ is the model's top-1 predicted label among the retained classes, and the top-2 probability gap is

$$\mathrm{gap}(p) = p_{(1)} - p_{(2)}$$

($p_{(1)} \geq p_{(2)}$ are the largest two entries of p).

We instantiate a membership attacker as a scoring function

$$f(x; \phi) \in [0, 1],$$

which takes the feature vector $\phi(x)$ as input and outputs a membership score (higher values indicating that x is more likely to be a training example). Concretely, $f(x; \phi)$ is implemented as a logistic-regression classifier trained on shadow data to distinguish *member* (train) vs. *non-member* (test) examples using $\phi(x)$ as features.

We then report ROC–AUC on the target model under two filters:

$$\mathrm{AUC}_{\mathrm{ret}} \;=\; \text{ROC-AUC}\big(\{i : y_i \neq c\}\big), \qquad \mathrm{AUC}_c \;=\; \text{ROC-AUC}\big(\{i : y_i = c\}\big).$$

Lower ROC-AUC scores indicate a stronger privacy guarantee. In particular we desire $\mathrm{AUC}_c \approx 0.5$.

5.3 Datasets and Experimental Setup

Datasets. We evaluate on three widely used multi-class document classification benchmarks: 20 Newsgroups, AG News, and DBPedia-14. These datasets are standard in text classification research and offer complementary levels of difficulty and class granularity, making them well-suited for studying class-level unlearning.

Specifically, 20 Newsgroups contains documents from 20 balanced topics spanning technology, politics, recreation, and science. Its relatively high number of classes and moderate document length make it a challenging test of whether unlearning preserves fine-grained decision boundaries.

AG News includes news articles labeled into 4 broad categories. Its lower intrinsic dimensionality allows us to test whether our method remains stable when classes are coarse-grained and semantically distinct, a setting where random relabeling tends to be particularly harmful.

DBPedia is a 14-class ontology-based dataset derived from Wikipedia, with short, well-structured descriptions. It provides a clean, high-accuracy baseline to assess whether unlearning introduces noise.

Together, these datasets cover a spectrum of class counts, label semantics, document lengths, and difficulty levels. They form a natural testbed for class-level unlearning as one class changes the structure of the remaining label space in different ways across each dataset.

Model. Our backbone classifier is a multinomial logistic regression model with ℓ_2 regularization, trained on TF–IDF features (Fig 1). For each dataset, we tokenize, lowercase, remove stopwords, and construct TF–IDF representations using sublinear term frequency, `min_df=2`, and a maximum vocabulary size of 50K. The classifier is trained with L-BFGS until convergence with a gradient tolerance of 10^{-5} and regularization strength $C = 10.0$ unless otherwise noted. A validation split comprising 10% of the training data is used for early stopping. This simple linear-softmax pipeline achieves strong baseline accuracy while exposing the influence of curvature (Section 4) in a transparent manner.

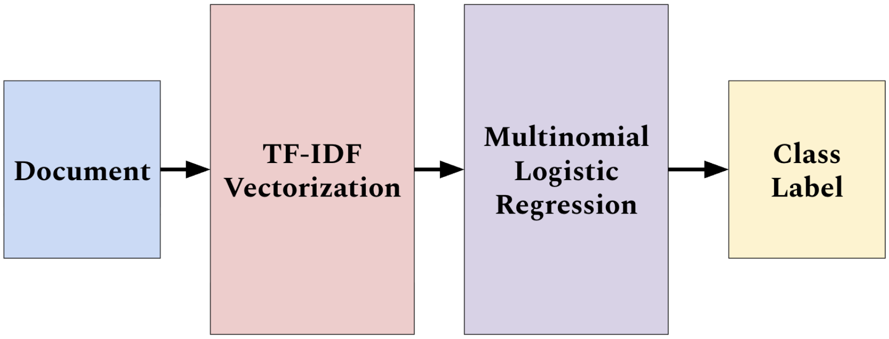


Figure 1: Architecture of the backbone document classifier.

Unlearning Procedure. For each class c, we compute the class-level gradient term for $\mathcal{D}_c$, solve the Hessian–vector system with conjugate gradients (tolerance 10^{-4}, maximum 200 iterations), apply the downweight update, and zero the deleted channel at inference. All experiments are repeated over 5 random seeds.

Attack Model. We adopt the shadow-model membership-inference setup used in prior work. We construct a pooled shadow dataset by repeatedly subsampling the original training corpus into disjoint *shadow_train*/*shadow_holdout* splits and training $S = 10$ shadow models, each replicating the full training pipeline. For post-unlearning evaluation, each shadow applies the same unlearning procedure (remove class c, apply the Hessian downweight, and zero the removed logit) before emitting prediction vectors.

For each example, we compute the attacker feature vector $\phi(x)$ defined in Section 5.2.2, and train a logistic-regression attacker on the pooled shadow features with balanced classes. The attacker is then evaluated on the target model to obtain $\mathrm{AUC}_{\mathrm{ret}}$ and AUC_c.

6 Results

6.1 Model Utility

We measure utility on the following conditions in Table 1:

(a) Overall accuracy on the pre-unlearning model

(b) Post-unlearning accuracy on the *retain classes* with **Golden Standard**

(c) Post-unlearning accuracy on the *retain classes* with **Random Relabeling**

(d) Post-unlearning accuracy on the *retain classes* with **Hessian Reassignment**

(e) The agreement rate on test items with $y{=}c$ between the **Golden Standard** and **Hessian Reassignment**, i.e., the fraction of deleted-class test documents for which both methods predict the same label among the retained classes.

Table 1: Utility comparison and agreement on deleted class.

Method	20 Newsgroup	AG News	DBPedia-14
Pre-Unlearning	93.58%	92.14%	98.33%
Golden Standard	93.73%	95.65%	98.43%
Random Relabel	83.45%	84.58%	88.46%
Hessian Reassignment	93.15%	95.04%	97.55%
Unlearned Class Agreement	87.77%	88.16%	90.46%

Across all three datasets, **Hessian Reassignment** matches the **Golden Standard** on retained-class accuracy while substantially outperforming **Random Relabeling**. The agreement on the deleted class is high (≈88%), indicating that the top-1 reassignment used in **Hessian Reassignment** is closely mirrored by the **Golden Standard**.

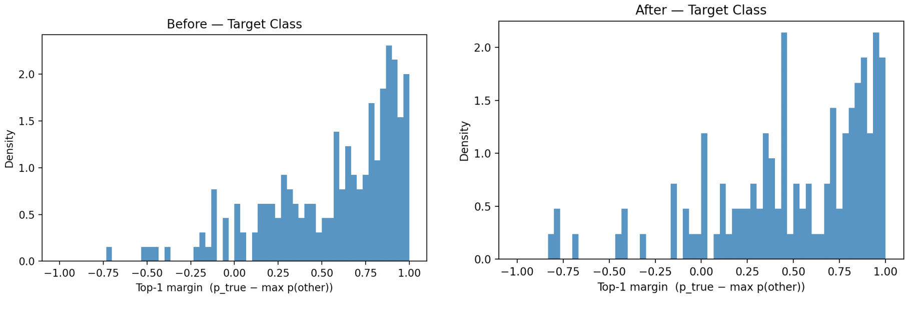
6.2 Confidence Distribution

To understand how unlearning affects the model's behavior, we study changes in *confidence*. For a classifier with output probabilities $p(x)$, the confidence that the model assigns to a label y is captured by the *top-1 probability margin*

$$m(x; y) \;=\; p_y(x) \;-\; \max_{j \neq y} p_j(x).$$

A positive margin means the model prefers y over all alternatives (**confident correct prediction**); a negative margin means some other label has higher probability (**confident misprediction**). Values near 1 correspond to near-total confidence, values near 0 to uncertainty, and values near -1 to high confidence in a wrong label.

Target-class documents. We first examine test examples whose original label is the deleted class c. Before unlearning (Fig. 2a), these examples have margins $m(x; c)$ that are mostly large and positive, indicating that the model strongly prefers class c on inputs it has been trained to recognize. After unlearning (Fig. 2b), we treat the Golden Standard reassignment $\tilde{y}(x)$ (obtained from retraining on $\mathcal{D}_{-c}$) as the reference label and plot $m(x; \tilde{y}(x))$. The post-unlearning distribution again shows large positive margins: **the model now assigns high confidence to the next-top-1 labels in $\mathcal{Y}_{-c}$ rather than to the removed class**.



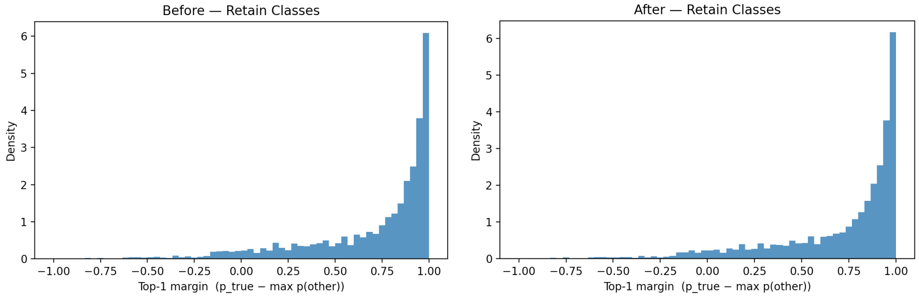
(a) Before Unlearning

(b) After Unlearning

Figure 2: Confidence Distribution on Target Class

Retained-class documents We next consider test examples with labels in the retained set $\mathcal{Y}_{-c}$. For these, we use the true label as reference and plot

$m(x; y)$ before and after unlearning (Fig. 3). In both cases, margins are sharply concentrated near 1.0. **The model remains highly confident and correct on retained classes**. Visually, the two curves are almost indistinguishable. Quantitatively, a Kolmogorov–Smirnov test on the before/after distributions yields $D = 0.0056$**,** $p = 1.00$, indicating no measurable statistical distributional shift after unlearning. The update therefore targets the removed class without disturbing how confident the model is on the remaining labels.



(a) Before Unlearning

(b) After Unlearning

Figure 3: Confidence Distribution on Retain Classes

6.3 Model Privacy

We evaluate membership-inference (MIA) risk with the fixed shadow-model attacker as described in Section 5.2.1. For each method, we report ROC–AUC on (i) documents whose labels are in the retained set ($\mathrm{AUC}_{\mathrm{ret}}$) and (ii) documents whose true label equals the removed class (AUC_c). Table 2 summarizes results across datasets.

Table 2: Membership-inference ROC–AUC.

Method	**Group**	**20Newsgroups**	**AG News**	**DBPedia**
Pre- Unlearning	Retain Target	0.6635 0.6946	0.5569 0.5510	0.6832 0.6794
Random Relabeling	Retain Target	0.6346 0.5530	0.5163 0.5181	0.6147 0.5228
Hessian Reassignment	Retain Target	0.6219 0.5109	0.5235 0.5161	0.6234 0.5094

In this setting, an ROC–AUC of 1.0 corresponds to a *perfect* attacker that can always distinguish training examples from non-members, and an ROC–AUC of 0.5 corresponds to a *random-guess* attacker that does no better than flipping a fair coin. Values closer to 0.5 therefore indicate stronger privacy, while values well above 0.5 indicate leakage.

Across all three datasets, **Hessian Reassignment** consistently drives the *Target AUC* (deleted class) toward the random-guess regime of 0.5. AUC_c drops from roughly 0.69 (pre-unlearning) and 0.65 (Random Relabeling) on 20 Newsgroups to 0.51 with our method, and shows similar behavior on AG News and DBPedia–14. At the same time, the *Retain AUC* (non-deleted classes) remains in the same range as the baselines. This pattern means that the Hessian downweight step is particularly effective at erasing membership information tied to the deleted class, without materially harming the privacy of the retained classes.

6.4 Noise Trade-Offs

We study the privacy-utility trade-off by injecting Gaussian logit noise at release after our Hessian downweight. For each regularization C and privacy target τ (desired AUC_{ret}), we search the smallest σ such that $\mathrm{AUC}_{ret} \le \tau$, and we report (i) retained-class accuracy at σ and (ii) $\log_{10} \sigma$.

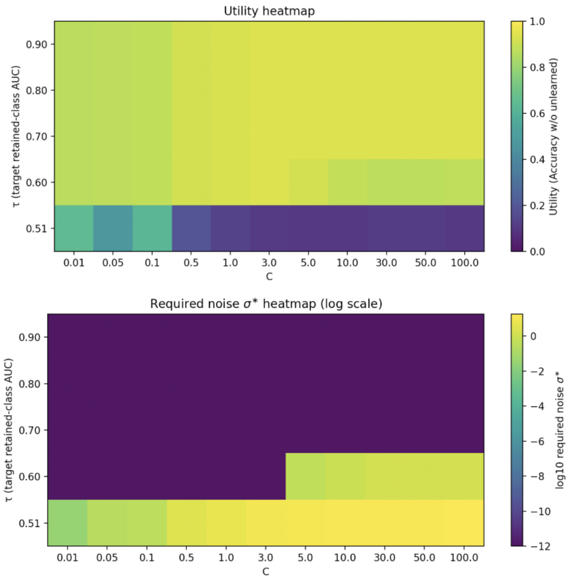


Figure 4: Privacy-utility heatmaps. *Top:* retained-class accuracy attained by the minimum noise σ that achieves the retain-class $\mathrm{AUC}_{ret} \le \tau$. *Bottom:* $\log_{10} \sigma$ (purple ≈ 0 noise).

Figure 4 illustrates our results. For moderate τ (e.g., 0.6), $\sigma \approx 0$ across most C and utility remains maximal. Pushing τ toward 0.5 requires nonzero noise chiefly at larger C, with a measured but localized utility drop. In all other settings, utility is essentially unchanged, indicating that our method already yields strong privacy on the deleted class without sacrificing performance on retained classes.

6.5 Model Efficiency

We measured end-to-end time to produce the released model after a deletion request for both full retraining (**Gold Standard**) and **Hessian Reassignment**.

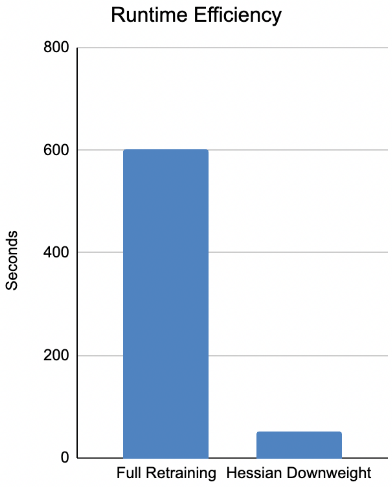


Figure 5: Runtime efficiency

Figure 5 compares the time to produce the released model on DPBedia-14. *Complete Retraining* takes roughly ∼602 seconds (around 10 minutes), whereas **Hessian Reassignment** finishes in about ∼53 seconds. The gain comes from replacing full optimization over the corpus with a single conjugate–gradient solve using Hessian–vector products. Importantly, this acceleration does not sacrifice quality. Sections 6.1 and 6.3 show retained–class accuracy and target class privacy remaining high.

7 Discussion

This work shows that deterministic next–top-1 reassignment through a single *Hessian downweight* step provides a practical route to class-level unlearning for document classifiers. Empirically, **Hessian Reassignment** matches **Golden Standard** (complete retraining) on retained-class utility and consistently surpasses **Random Relabeling**. The retained-class confidence distribution remains effectively unchanged (e.g., KS p=1.0 on 20 Newsgroups), while the model is confident in its next top-1 prediction for the target class. Privacy follows suit: membership-inference ROC–AUC on the target class moves toward chance ($\text{AUC}_c \approx 0.5$) without sacrificing retained-class accuracy.

The top–1 reassignment preserves the model's *relative* score ordering among non-target labels, avoiding contradictory supervision that random relabeling introduces. The Hessian downweight then acts as an inexact Newton correction, canceling the deleted class's first-order influence using the local curvature, yielding a parameter move that respects the existing geometry of the decision boundaries. As a result, no iterative retraining is required.

For multi-class text settings with moderate regularization, the method is a strong default: it achieves near–retraining utility on retained classes and drives the target-class MIA toward 0.5 without extra noise. When AUC_{ret} remains above a policy threshold, our heatmaps indicate that only small logit noise is needed to close the gap.

8 Future Work

Looking ahead, several directions appear promising. First, the Hessian downweight step could be adapted to richer text pipelines, including neural networks and transformer models. Additionally, developing lightweight certificates that quantify how much influence remains after the downweight update would move the approach closer to provable guarantees. Together, these directions would help translate the simplicity of our framework to a wider range of modern NLP systems.

9 Conclusion

We presented a simple, efficient algorithm for class-level unlearning in document classification: deterministically reassign deleted-class examples to their next top–1 label and apply a single *Hessian downweight* to remove the deleted class's first-order influence. Across standard text benchmarks, **Hessian Reassignment** matches the **Golden Standard** (complete retraining) on retained-class accuracy, substantially outperforms **Random Relabeling**, and drives target-class membership–inference AUC toward chance, all while leaving retained-class confidence distributions effectively unchanged.

Conceptually, next–top-1 reassignment preserves the model's learned ordering among the remaining labels, and the downweight step uses local curvature to make the smallest corrective move consistent with that structure. Practically, this yields a training-free alternative to complete retraining with strong utility–privacy balance.

10 Acknowledgement

I would like to thank the MIT PRIMES program for their support and the opportunity to conduct this research. I am also grateful to my mentor, Mayuri

Sridhar, for her extensive guidance and help.

References

[1] S. Shastri, M. Wasserman, and V. Chidambaram. The seven sins of personal-data processing systems under GDPR. *USENIX HotCloud*. 2019.

[2] Bill text. https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=201720180AB375. 2024.

[3] General Data Protection Regulation. https://gdpr-info.eu/. 2018.

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

[5] L. Bourtoule, M. Chandrasekaran, S. Choquette-Choo, et al. Machine Unlearning. In *IEEE Symposium on Security and Privacy*, 2021.

[6] Ginart, A., Guan, M., Valiant, G., Zou, J. Making AI forget you: Data deletion in machine learning. NeurIPS. *arXiv preprint arXiv:1907.05012*, 2019.

[7] Neel, S., Roth, A., Sharifi-Malvajerdi, S. Descent-to-Delete: Gradient-based methods for machine unlearning. *arXiv preprint arXiv:2007.02923*, 2021.

[8] P. W. Koh and P. Liang. Understanding Black-box Predictions via Influence Functions. In *ICML*, 2017.

[9] L. Kang, M. A. Souibgui, F. Yang, L. Gomez, E. Valveny, and D. Karatzas, "Machine Unlearning for Document Classification," ICDAR, 2024.

[10] S. B. R. Chowdhury, J. Steinhardt, and M. Hardt, "Towards Scalable Exact Machine Unlearning," ICLR, 2025.

[11] A. Warnecke, S. Jha, A. Machanavajjhala, and M. Backes, "Machine Unlearning of Features and Labels," NDSS, 2023.

[12] Y. Allouah, T. Kerdoncuff, and F. Bach, "The Utility and Complexity of in- and out-of-Distribution Machine Unlearning" *arXiv preprint arXiv:2412.09119*, 2025.

[13] M. Xu, P. Kairouz, and B. Raj, "Machine Unlearning: Challenges in Data Quality and Access," IJCAI, 2024.

[14] M. R. U. Rashid, M. T. Rahman, and P. Liu, "Leveraging Machine-Unlearning on Pretrained Language Models for Privacy-Preserving Fine-Tuning," AAAI, 2025.

[15] A. Deeb and F. Roger, "Do Unlearning Methods Remove Information from Neural Networks?" ICLR, 2025.

[16] Y. Zhao, L. Du, X. Ding, K. Xiong, Z. Sun, J. Shi, T. Liu, B. Qin, "Deciphering the Impact of Pretraining Data on Large Language Models via Machine Unlearning," Findings of ACL, 2024.

[17] G. Pruthi, F. Liu, S. Kale, and M. Sundararajan, "Estimating Training Data Influence by Tracing Gradient Descent (TracIn)," NeurIPS, 2020.

[18] S. Basu, X. You, and S. Feizi, "On Second-Order Group Influence Functions for Black-Box Predictions," ICML, 2020.

[19] S. Basu, P. Pope, and S. Feizi, "Influence Functions in Deep Learning Are Fragile," ICML, 2021.

[20] Z. Hammoudeh and D. Lowd, "Training Data Influence Analysis and Estimation: A Survey," Machine Learning, 2024.

[21] M. Isonuma, J. Kawakami, and J. Suzuki, "Unlearning Traces the Influential Training Data of Language Models," ACL, 2024.

[22] H. Tan, S. Wu, B. Zhao, Z. Xie, X. Qi, "Diff-In: Data Influence Estimation with Differential Approximation" ICLR 2025.

[23] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership Inference Attacks Against Machine Learning Models," IEEE Symposium on Security and Privacy, 2017.

[24] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, and M. Backes, "ML-Leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models," NDSS, 2019.

[25] S. Yeom, M. Fredrikson, and S. Jha, "Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting," IEEE CSF, 2018.

[26] Z. Zhang, C. Yan, B. A. Malin, "Membership Inference Attacks Against Synthetic Health Data," npj Digital Medicine, 2021.

[27] J. Chen, D. Yang, "Unlearn What You Want to Forget: Efficient Unlearning for LMs," EMNLP, 2023.

[28] J. Yao, E. Chien, M. Du, X. Niu, T. Wang, Z. Cheng, X. Yue, "Machine Unlearning of Pre-Trained Large Language Models," *arXiv preprint arXiv:2402.15159*, 2024.

[29] Z. He, T. Li, X. Cheng, Z. Huang, X. Huang, "Towards Natural Machine Unlearning," IEEE TPAMI, 2025.

[30] J. Li and S. Ghosh, "Random Relabeling for Efficient Machine Unlearning," *arXiv preprint arXiv:2305.12320*, 2023.