PALMU: REGENERATING SUPERVISION FOR MACHINE UNLEARNING IN SELF-TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Self-training fits a labeling model on a small labeled dataset and uses its predictions to assign pseudo-labels and select unlabeled examples for downstream training. After deletion of a labeled record, its influence may remain in this generated supervision. Removing the record from downstream training while keeping stored pseudo labels and selection decisions can therefore fail to reproduce full retraining without that record. We propose Pseudo-label Aware Labeled data Machine Unlearning (PALMU). PALMU updates the labeling model on retained labeled data and regenerates pseudo labels for the unlabeled pool. Examples whose predicted class probability meets the confidence threshold enter downstream training. PALMU then updates the downstream classifier without fitting both classifiers to convergence. For fixed representations, we bound PALMU's distance from full retraining using remaining downstream optimization error and supervision differences from the fully retrained labeling model. Across four datasets at deletion ratios of 5%, 10%, 20%, and 30%, PALMU meets predefined agreement and utility criteria in 397 of 400 fixed representation requests. Its mean predictive total variation is 4.3 to 56.6 times lower than the closest baseline evaluated against the same retraining reference. On CommitBERT, it retains 1.56 to 1.83 macro F1 points over labeled-only retraining. For MLPs with learned representations, updating only the final classifier does not match retraining.

1 INTRODUCTION

ML models trained on user data have become critical to education, medicine, and other domains. The expansion of these models carry with them obligations to safeguard user data, ensure long-term privacy, and to respond to legal mandates for deletion under a *right to be forgotten* (Kostopoulos & Kotsiantis, 2021; Jiao et al., 2024; Juliussen et al., 2023; Nguyen et al., 2025). *Full retraining* of models to guarantee compliance is expensive thus motivating work on machine unlearning with the goal of achieving the same effect as full retraining at a fraction of the cost (Bourtoule et al., 2021; Golatkar et al., 2020; Kurmanji et al., 2023). These approaches have shown some success in unlearning domains (Chourasia & Shah, 2023; Pan et al., 2022; Wang et al., 2024) However not all ML approaches lend themselves to effective unlearning. This is particularly true of semi-supervised and self-training models which are increasingly common in privacy-sensitive domains (Kostopoulos & Kotsiantis, 2021; Jiao et al., 2024; Song et al., 2024).

Self-training creates an additional deletion pathway. A *labeling model* is fitted on a small labeled set and assigns pseudo-labels and confidence scores to a larger unlabeled pool (Sohn et al., 2020; Arazo et al., 2020). A *downstream classifier* is then fitted on the labeled examples and the accepted pseudo-labeled examples. Deleting one labeled record therefore changes more than its own loss. It can change the labeling model which can then change the supervision assigned to remaining unlabeled examples. We call the resulting difference a *deletion-induced supervision change*. The goal is to match the pseudo-labels and example selection decisions produced by repeating self-training on the retained labeled data.

We propose PALMU which approximates the full retraining procedure for self-training without fitting both classifiers to convergence. Our implementation first updates the downstream classifier using retained labels and stored pseudo-labels. PALMU then updates the labeling model on the retained labeled data, regenerates pseudo-labels and acceptance decisions over the complete unlabeled

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.

pool, and updates the downstream classifier while keeping this regenerative supervision fixed. When a pseudo-label or acceptance decision changes, the new objective removes the old loss contribution and adds its replacement. However, the preliminary downstream update can be omitted, as shown by the comparison using the same number of downstream updates in Section 5.5. We therefore define pseudo-label regeneration and downstream correction as the core operations of PALMU.

Our experiments show that deletion-induced supervision changes matter in practice. Across four datasets with deep fixed representations, PALMU achieves a mean predictive total variation from full retraining between 0.0003 and 0.0011 at 10% deletion. The closest baseline results range from 0.016 to 0.036. PALMU therefore reproduces the predictions of full retraining at least one order of magnitude more closely than the evaluated alternatives.

Agreement with full retraining does not necessarily imply higher test performance. In a focused CommitBERT study, regeneration moves the downstream classifier closer to full retraining in all 40 deletion requests, but macro-F1 decreases relative to the preliminary update in 39 requests. A matched computation comparison also shows that directly updating the downstream classifier on regenerated supervision is faster and closer to full retraining than performing the two updates in sequence. The evidence therefore supports regeneration as the essential correction. It does not support requiring the preliminary downstream classifier update.

Our primary claim concerned fixed representation. In this setting, the encoder or input representation is chosen independently of fitting on the downstream labeled set, and only the two regularized linear classifiers depend on deletion-affected data. This covers the frozen BERT, ResNet-50, and Llama encoders evaluated here. It also makes the classifier objectives convex, permitting efficient second-order updates and direct comparison with full retraining. This setting is sufficient to isolate the generated-supervision problem, but it does not show that a head-only correction extends to learned representations. The MLP experiments examine this boundary because their hidden layers are themselves fitted on deletion-affected data.

Contributions.

- We formalize labeled-data deletion in one-round self-training, including its effect on pseudo-label assignments and confidence-based acceptance decisions.
- We develop PALMU, which updates the labeling model, regenerates supervision over the unlabeled pool, and efficiently corrects the downstream classifier under fixed representations. A distance-to-retraining bound separates incomplete optimization from disagreement in regenerated supervision.
- Across four datasets, we identify when regeneration improve agreement with full retraining, when pseudo-labels preserve predictive utility, and where a head-only correction fails because the representation was learned from deletion-affected data.

2 Related Work

Machine unlearning can limit retraining work through sharding or update a trained model using curvature, distillation, saliency, or preference objectives (Bourtoule et al., 2021; Golatkar et al., 2020; Kurmanji et al., 2023; Fan et al., 2023; Zhang et al., 2024; Foster et al., 2024). To compare these approaches in self-training, we evaluate classifier adaptations with stored supervision and, where applicable, with separately regenerated supervision. The comparison tests whether updating the classifier is enough when deleting a labeled example can also change the supervision assigned to the unlabeled pool.

Related work studies unlearning with partial, missing, or weak labels (Tang et al., 2024; Shen et al., 2024; Peng et al., 2025; Zhu et al., 2026). DMM uses data attribution to estimate the predictions of a retrained model and then fits the unlearned model to those estimates (Georgiev et al., 2025). Representation misdirection and simplified preference methods target forgetting in large models (Li et al., 2024; Fan et al., 2026), and benchmarks such as TOFU and MUSE measure that forgetting (Maini et al., 2024; Shi et al., 2025). Our question concerns labels generated by a model fitted on data that included the deleted examples. Confidence-filtered pseudo-labeling makes both the assigned class and the decision to accept an example depend on that model (Arazo et al., 2020; Sohn et al., 2020; Zhang et al., 2021). Full self-training retraining is our reference because it recomputes both after deletion.

3 PROBLEM SETUP

Data and representations. We study machine unlearning in the semi-supervised setting, where a labeled dataset drives self-training over a larger unlabeled pool. Let $\mathcal{D}_\ell = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n_\ell}$ be the labeled dataset with labels in $\{1, \dots, C\}$, and let $\mathcal{D}_u = \{\mathbf{x}_j\}_{j=1}^{n_u}$ be a disjoint unlabeled pool. In our fixed representation setting, the labeling model and downstream classifier share a feature map $\mathbf{h}\colon \mathcal{X} \to \mathbb{R}^d$, with a constant coordinate for the intercept. This map converts each input into a feature vector and is chosen before either model is fitted. It remains unchanged under a deletion request. The later MLP experiments test a setting outside this assumption because their hidden representations are learned from the labeled data.

Both classifiers are linear softmax heads over $\mathbf{h}$. For head parameters $\mathbf{v} = (\mathbf{v}_1, \dots, \mathbf{v}_C)$, the probability of class c is $p_c(\mathbf{x}; \mathbf{v}) = \frac{\exp(\mathbf{v}_c^\top \mathbf{h}(\mathbf{x}))}{\sum_{k=1}^{C} \exp(\mathbf{v}_k^\top \mathbf{h}(\mathbf{x}))}$. The labeling classifier uses parameters $\boldsymbol{\theta}$ and the downstream classifier uses parameters $\mathbf{w}$, and we write $\boldsymbol{\theta}_0$ and $\mathbf{w}_0$ for the two fitted on the original labeled dataset. Given $\boldsymbol{\theta}$, the generated label and acceptance indicator for pool example $\mathbf{x}_j$ are $\hat{y}_j(\boldsymbol{\theta}) = \arg\max_c p_c(\mathbf{x}_j; \boldsymbol{\theta})$ and $a_j(\boldsymbol{\theta}) = \mathbf{1}\{\max_c p_c(\mathbf{x}_j; \boldsymbol{\theta}) \geq \tau\}$, where $\tau \in (0, 1)$ is the confidence threshold and $a_j(\boldsymbol{\theta}) = 1$ admits the example into downstream training as $\hat{\mathcal{D}}_u(\boldsymbol{\theta}) = \{(\mathbf{x}_j, \hat{y}_j(\boldsymbol{\theta})) : a_j(\boldsymbol{\theta}) = 1\}$. Changing the labeling classifier can change the assigned class, the acceptance decision, or both.

Labeling and downstream objective. Write $\mathrm{CE}(\mathbf{v}; \mathbf{x}, y) = -\log p_y(\mathbf{x}; \mathbf{v})$ for multiclass cross-entropy, and let $\mathcal{S} \subseteq \mathcal{D}_\ell$ be the nonempty labeled subset used in a fit, which lets the same objectives apply before and after deletion. The labeling objective fits $\boldsymbol{\theta}$ on the observed labels in $\mathcal{S}$,

$$\mathcal{L}_{\mathrm{lab}}(\boldsymbol{\theta}; \mathcal{S}) = \frac{1}{|\mathcal{S}|} \sum_{(\mathbf{x}, y) \in \mathcal{S}} \mathrm{CE}(\boldsymbol{\theta}; \mathbf{x}, y) + \frac{\lambda_\ell}{2} \|\boldsymbol{\theta}\|_F^2 . \tag{1}$$

The downstream objective fits $\mathbf{w}$ on the observed labels together with the pseudo-labels accepted by $\boldsymbol{\theta}$, $\mathcal{L}_{\mathrm{self}}(\mathbf{w}; \mathcal{S}, \boldsymbol{\theta}) = \frac{1}{|\mathcal{S}|} \sum_{(\mathbf{x}, y) \in \mathcal{S}} \mathrm{CE}(\mathbf{w}; \mathbf{x}, y) + \frac{\alpha}{n_u} \sum_{j=1}^{n_u} a_j(\boldsymbol{\theta})\, \mathrm{CE}\big(\mathbf{w}; \mathbf{x}_j, \hat{y}_j(\boldsymbol{\theta})\big) + \frac{\lambda}{2} \|\mathbf{w}\|_F^2$. The coefficient α controls the generated supervision, λ_ℓ and λ regularize the two classifiers, and the pool term is normalized by the original pool size n_u so changing the number of accepted examples does not rescale the rest. The original procedure minimizes $\mathcal{L}_{\mathrm{lab}}(\boldsymbol{\theta}; \mathcal{D}_\ell)$ for $\boldsymbol{\theta}_0$, generates supervision over $\mathcal{D}_u$, and minimizes $\mathcal{L}_{\mathrm{self}}(\mathbf{w}; \mathcal{D}_\ell, \boldsymbol{\theta}_0)$ for $\mathbf{w}_0$, the one-round self-training procedure considered here.

The forget request. Given a forget set $\mathcal{D}_f \subset \mathcal{D}_\ell$, the retained set is $\mathcal{D}_r = \mathcal{D}_\ell \setminus \mathcal{D}_f$. Full retraining minimizes $\mathcal{L}_{\mathrm{lab}}(\boldsymbol{\theta}; \mathcal{D}_r)$ for $\boldsymbol{\theta}_{\mathrm{rt}}$, regenerates all pseudo-labels and acceptance decisions over $\mathcal{D}_u$, and minimizes $\mathcal{L}_{\mathrm{self}}(\mathbf{w}; \mathcal{D}_r, \boldsymbol{\theta}_{\mathrm{rt}})$ for $\mathbf{w}_{\mathrm{rt}}$, with the unlabeled dataset $\mathcal{D}_u$ and the selected hyperparameters held fixed. This removes the forget set from both the observed-label term and the model that generates supervision. Keeping the original pseudo-labels while updating only the downstream classifier on $\mathcal{D}_r$ removes the direct contribution of $\mathcal{D}_f$ but retains its influence through supervision generated by $\boldsymbol{\theta}_0$, so we evaluate that stored-supervision procedure as a control in Stage 1 and the baselines rather than as the full-retraining reference.

Evaluation setup. Our evaluation asks whether the updated classifier retains predictive utility, whether it reproduces the predictions of full retraining, and how much time the deletion update takes. We measure utility with macro-F1 on a fixed test set, which averages the per-class F1 with equal weight so the majority class does not dominate, and it reflects task performance rather than agreement with full retraining. We measure agreement with predictive total variation (TV). For an updated classifier $\mathbf{w}$ with probability vector $p(\mathbf{x}; \mathbf{w})$ and the full-retraining classifier $\mathbf{w}_{\mathrm{rt}}$, over a test set of N examples, $\mathrm{TV}(\mathbf{w}, \mathbf{w}_{\mathrm{rt}}) = \frac{1}{2N} \sum_{i=1}^{N} \|p(\mathbf{x}_i; \mathbf{w}) - p(\mathbf{x}_i; \mathbf{w}_{\mathrm{rt}})\|_1$

where zero means identical class probabilities everywhere and larger values mean greater disagreement. Efficiency is the time to process a deletion request, covering the labeling update, pseudo-label regeneration, and the downstream update. As a secondary diagnostic we report the forgetting score $\mathrm{FS} = 1 - 2|\mathrm{AUC} - 0.5|$ from a loss-based membership inference attack, where one means chance and zero means perfect separation. Neither FS nor TV certifies that information about the deleted records has been removed, and TV measures agreement only on the evaluated test examples.

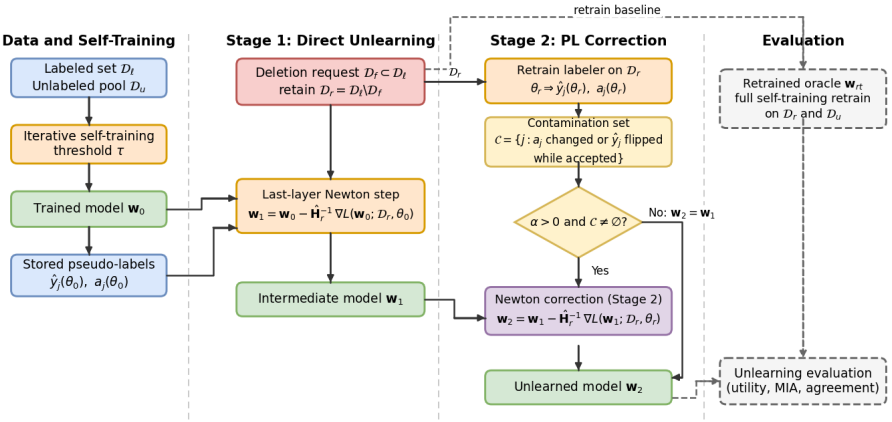

Figure 1: Overview of PALMU. Stage 1 updates $\mathbf{w}_0$ on retained labels and stored pseudo-labels to obtain $\mathbf{w}_1$. Stage 2 updates the initial labeling classifier using $\mathcal{D}_r$, regenerates pseudo-labels and acceptance on $\mathcal{D}_u$, and corrects the final model to obtain $\mathbf{w}_2$. We skip the extra correction when no active pseudo-label's contribution changes. The dashed retraining branch is used only for evaluation.

4 PALMU: Pseudo-Label-Aware Labeled Machine Unlearning

The forget set $\mathcal{D}_f$ affects the self-training objective in two ways. Its direct effect comes from the labeled loss terms associated with examples in $\mathcal{D}_f$. Its indirect effect arises because the original labeling classifier $\boldsymbol{\theta}_0$ was fitted on $\mathcal{D}_\ell$, which included $\mathcal{D}_f$, and was then used to generate pseudo-labels and acceptance decisions for $\mathcal{D}_u$. Stage 1 updates the downstream classifier using $\mathcal{D}_r$ but retains this original generated supervision. Stage 2 updates the labeling classifier using $\mathcal{D}_r$, regenerates supervision, and then corrects the downstream classifier on the resulting objective. Figure 1 shows the two stages. The Newton budgets K_1, K_ℓ, and K_2 are set on development data and fixed before evaluation, all updates act on the classifier heads, and the fixed representation $\mathbf{h}(\mathbf{x})$ from Section 3 is unchanged.

4.1 Stage 1: Direct Unlearning via the Newton Step

Stage 1 starts from $\mathbf{w}_0$ and moves toward the downstream objective defined by the retained labeled set and the original generated supervision. Set $\mathbf{w}_1^{(0)} = \mathbf{w}_0$. For $t = 0, \ldots, K_1 - 1$ denote the number of Stage 1 downstream updates, define At iteration t, let $\mathbf{g}_1^{(t)} = \nabla_{\mathbf{w}} \mathcal{L}_{\text{self}}(\mathbf{w}_1^{(t)}; \mathcal{D}_r, \boldsymbol{\theta}_0)$ and $\mathbf{H}_1^{(t)} = \nabla_{\mathbf{w}}^2 \mathcal{L}_{\text{self}}(\mathbf{w}_1^{(t)}; \mathcal{D}_r, \boldsymbol{\theta}_0)$. Each Newton update solves

$$\mathbf{H}_1^{(t)} \mathbf{v}_1^{(t)} = \mathbf{g}_1^{(t)}, \qquad \mathbf{w}_1^{(t+1)} = \mathbf{w}_1^{(t)} - \eta_1^{(t)} \mathbf{v}_1^{(t)}. \tag{2}$$

After K_1 updates, we set $\mathbf{w}_1 = \mathbf{w}_1^{(K_1)}$. The implementation computes the Newton direction by solving the linear system $\mathbf{H}\mathbf{v} = \mathbf{g}$ using conjugate gradients and Hessian vector products, without explicitly constructing the Hessian or its inverse. The budget K_1 is selected using development data. The step size uses Armijo backtracking (Armijo, 1966), and each Newton system is solved with preconditioned conjugate gradients on Hessian-vector products Nocedal & Wright, 2006, so the coupled softmax Hessian is never formed or inverted. Appendix B gives the step-size rule, the Hessian product, and the stopping conditions.

4.2 Stage 2: Generated-Supervision Correction

Stage 1 still uses supervision generated by $\boldsymbol{\theta}_0$, which was fitted before deletion. Stage 2 performs three operation in order. It first applies K_ℓ Newton updates to $\mathcal{L}_{\text{lab}}(\boldsymbol{\theta}; \mathcal{D}_r)$, starting from $\boldsymbol{\theta}_0$, to obtain $\boldsymbol{\theta}_r$. These updates use the same linear solver and line search described above. It then recomputes $\hat{y}_j(\boldsymbol{\theta}_r)$ and $a_j(\boldsymbol{\theta}_r)$ for every example in $\mathcal{D}_u$. The confidence threshold τ is kept fixed, so any

change in acceptance comes from the updated labeling classifier rather than a new threshold. The downstream correction begins only after this regeneration is complete.

Let $\mathcal{C}$ contain the pool indices whose active contributions change. An unlabeled example's index belongs to $\mathcal{C}$ if its acceptance decision changes, or if it is accepted both before and after regeneration but receives a different pseudo label. If $\alpha > 0$ and $\mathcal{C} \neq \varnothing$, define $\mathbf{g}_2 = \nabla_{\mathbf{w}} \mathcal{L}_{\text{self}}(\mathbf{w}_1; \mathcal{D}_r, \boldsymbol{\theta}_r)$ and $\mathbf{H}_2 = \nabla^2_{\mathbf{w}} \mathcal{L}_{\text{self}}(\mathbf{w}_1; \mathcal{D}_r, \boldsymbol{\theta}_r)$. One downstream correction is

$$\mathbf{H}_2 \mathbf{v}_2 = \mathbf{g}_2, \qquad \mathbf{w}_2 = \mathbf{w}_1 - \eta_2 \mathbf{v}_2, \tag{3}$$

where η_2 is chosen by the same Armijo rule. If the selected budget K_2 exceeds one, all quantities are recomputed after every update. If $\alpha = 0$ or $\mathcal{C} = \varnothing$, the downstream objective has no changed pseudo-label contribution and we set $\mathbf{w}_2 = \mathbf{w}_1$.

Updates equation 2 and equation 3 show one step each. When the selected budget exceeds one, the gradient, Hessian, and line search are recomputed at each step.

Let $\mathbf{e}_y \in \mathbb{R}^C$ denote the one-hot vector for class y. For an accepted pool example with assigned label y, its contribution to the downstream cross-entropy gradient is $\frac{\alpha}{n_u} \mathbf{h}(\mathbf{x}_j) \left(p(\mathbf{x}_j; \mathbf{w}) - \mathbf{e}_y\right)^\top$. Consider an example accepted before and after regeneration, meaning $a_j(\boldsymbol{\theta}_0) = a_j(\boldsymbol{\theta}_r) = 1$. If its pseudo-label changes, subtracting the stored gradient contribution from the regenerated contribution gives $\frac{\alpha}{n_u} \mathbf{h}(\mathbf{x}_j) \left(\mathbf{e}_{\hat{y}_j(\boldsymbol{\theta}_0)} - \mathbf{e}_{\hat{y}_j(\boldsymbol{\theta}_r)}\right)^\top$,

The probability vector cancels because both contributions are evaluated at the same downstream parameters $\mathbf{w}$. If the acceptance decision changes, the regenerated objective adds or removes the complete pseudo-label loss for that example. Keeping the stored contribution while adding the regenerated one would optimize a different objective.

Distance from full retraining. For fixed representations, Proposition 1 in Appendix A bounds PALMU's parameter distance from full retraining using remaining downstream optimization error and disagreement between regenerated supervision and supervision produced by the fully retrained labeling model. Further downstream optimization alone need not eliminate the discrepancy when these sources of supervision differ. Section 5.4 examines the two error sources experimentally. The bound depends on the retraining reference, can be loose when λ is small, and does not certify deletion.

5 Experiments

Our experiments test whether PALMU approximates full retraining with fixed representations, how updating generated supervision improves that agreement, and where the method fails when representations are learned from data that include the deleted records.

5.1 Experimental Setup

Datasets and requests. We evaluate CommitBERT and EDM (commit-message classification), AG News (news classification), and CIFAR-10 (image classification) (Jung, 2021; Gitinabard et al., 2023; Zhang et al., 2015; Krizhevsky et al., 2009) at forget ratios of 5%, 10%, 20%, and 30%. Each setting uses ten paired deletion requests. Within each request, methods share the forget set, original model, and unlabeled pool. Hyperparameters are selected using validation data and separate tuning requests at 10% deletion, then held fixed across evaluation requests and ratios. The confirmation and ablation use separate calibrations and request seeds (Appendix D).

Representations. Fixed-representation experiments use pool-fitted TF-IDF or PCA features and frozen BERT, ResNet-50 (He et al., 2016), or Llama-3.1-8B (Grattafiori et al., 2024) features. MLPs instead learn hidden layers from labeled data, making their representations deletion-dependent. PALMU updates the labeling and downstream heads. Appendix C gives dataset sizes, architectures, and preprocessing.

References and baselines. Full retraining of the self training pipeline provides the reference for agreement. Retraining on labeled data alone tests whether generated supervision can be discarded.

Table 1: Deep fixed representations at 10% deletion. Entries are means over ten paired request seeds. BERT is used for the three text datasets and ResNet-50 for CIFAR-10. F1 is macro-F1 and TV is prediction discrepancy from full retraining. †SISA uses its own sharded reference, so its zero TV is not agreement with the unsharded model. Full results appear in Appendix H.

	CommitBERT		EDM		AG News		CIFAR-10	
Method	F1↑	TV↓	F1↑	TV↓	F1↑	TV↓	F1↑	TV↓
Full retraining	0.8475	0.00000	0.7983	0.00000	0.8637	0.00000	0.8711	0.00000
Labeled-only retraining	0.8339	0.07845	0.7973	0.00643	0.8614	0.05694	0.8701	0.02152
Fisher	0.8506	0.02391	0.8179	0.03831	0.8648	0.02086	0.8720	0.01792
Fisher + regeneration	0.8506	0.02391	0.8179	0.03831	0.8648	0.02086	0.8720	0.01792
SCRUB	0.8507	0.02296	0.8160	0.03574	0.8649	0.01980	0.8716	0.01610
SCRUB + regeneration	0.8505	0.02240	0.8160	0.03570	0.8653	0.01922	0.8716	0.01586
SalUn	0.8506	0.02390	0.8200	0.03747	0.8652	0.02080	0.8719	0.01774
SalUn + regeneration	0.8506	0.02389	0.8200	0.03747	0.8653	0.02079	0.8719	0.01774
SISA†	0.8397	0.00000	0.7855	0.00000	0.8608	0.00000	0.8685	0.00000
PALMU	0.8475	0.00113	0.7976	0.00034	0.8639	0.00107	0.8712	0.00034

We also compare Fisher, SCRUB, SalUn, SISA, and classification adaptations of NPO (Zhang et al., 2024), NPO+RT, and SSD (Foster et al., 2024). Fisher, SCRUB, and SalUn are evaluated with stored supervision and separately tuned regeneration wrappers. PALMU and all baselines except SISA use the same full retraining reference for self training. SISA uses a separate reference that retrains its sharded models (Appendix E).

Agreement criteria. We require TV at most 0.01 and a macro F1 drop from full retraining of at most 0.01. We distinguish admissible means from individual request passes. The supplement reports uncertainty, the membership attack diagnostic FS, and runtime. Timings include updates and regeneration on cached features but exclude feature extraction and evaluation. FS does not certify deletion, and request variation on previously used test sets does not constitute evaluation on new datasets.

5.2 Comparison Across Four Datasets

We first test whether PALMU reproduces full retraining across tasks while preserving test performance. Table 1 compares agreement with full retraining and test performance on deep fixed representations. PALMU is inside both mean tolerances on all four datasets, whereas the evaluated Fisher, SCRUB, and SalUn variants remain outside the TV tolerance. Their similar or higher macro F1 shows that test performance alone cannot establish agreement. Regeneration produces little improvement for these baseline configurations, while PALMU's combined correction achieves substantially closer agreement.

To separate fidelity from the utility of pseudo-labels, we compare with labeled-only retraining. PALMU retains about 1.36 macro F1 points over retraining on labeled data alone on CommitBERT. On AG News and CIFAR-10, small performance differences coexist with TV of 0.05694 and 0.02152 for that alternative. Similar accuracy therefore does not reproduce the self training reference. On EDM, retraining on labeled data alone already meets the TV tolerance. The utility of preserving pseudo labels is task dependent, while PALMU targets the specified retraining procedure. Appendix H reports full results.

5.3 Agreement with full retraining for recent methods

We next test whether recent unlearning methods close the agreement gap. Table 2 reports predictive TV for the eight fixed-representation settings at 10% deletion. PALMU has lower TV than NPO, NPO+RT, and SSD in every setting. The closest recent method has approximately 5 to 168 times PALMU's discrepancy from full retraining. This extends the agreement advantage to the evaluated recent methods under the same retraining reference. Their macro-F1 sits at or slightly above retraining on all eight settings. The disagreement therefore persists despite comparable test performance. These methods can preserve accuracy without reproducing the predictions of the retained self training pipeline. NPO and SSD each finish in a fraction of a second and are faster than PALMU on the

Table 2: Modern classification-adapted unlearning baselines at 10% deletion on the fixed-representation cells. Entries are mean predictive TV from full retraining over ten paired request seeds. The last column reports how much larger the nearest of the three modern methods is than PALMU. PALMU and the retraining reference are recomputed within this comparison. Full macro-F1, FS, and runtime appear in Appendix H.

Dataset	Encoder	**PALMU** TV↓	NPO TV↓	NPO+RT TV↓	SSD TV↓	Best modern over PALMU
CommitBERT	LR	**0.00006**	0.00954	0.00919	0.01181	147×
CommitBERT	Llama-3.1-8B	**0.00454**	0.02647	0.02417	0.02351	5×
EDM	LR	**0.00198**	0.04037	0.03818	0.04113	19×
EDM	Llama-3.1-8B	**0.00036**	0.05412	0.04799	0.03676	101×
AG News	LR	**0.00026**	0.04555	0.04426	0.05112	168×
AG News	BERT	**0.00170**	0.02829	0.02752	0.02771	16×
CIFAR-10	LR	**0.00178**	0.08004	0.07701	0.08674	43×
CIFAR-10	ResNet-50	**0.00047**	0.02254	0.02207	0.02279	47×

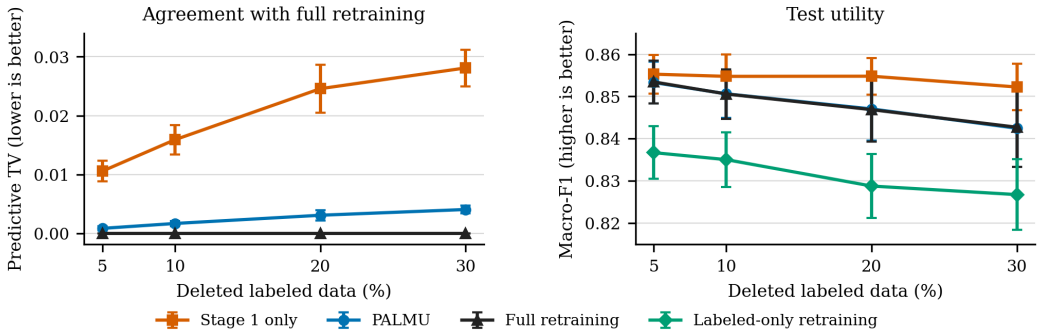

Figure 2: Stage 2 verification on CommitBERT/BERT across ten paired deletion requests. The left panel reports predictive TV from full self-training retraining, and the right panel reports test macro-F1. Points are means and error bars are sample standard deviations. Stage 2 improves agreement with full retraining despite a slight decrease in macro F1 relative to Stage 1. PALMU also retains an advantage over retraining on labeled data alone. Exact values and protocol details appear in Table 4.

cached deep features. Their speed provides a computational advantage, but their larger TV shows that this advantage comes with weaker agreement with full retraining. PALMU provides a closer approximation to the specified deletion reference.

The MLP results show the limit of this advantage. None of the evaluated methods meets the agreement tolerance. Plain NPO becomes unstable and loses accuracy on three of the four datasets, and NPO+RT together with SSD achieve TV similar to PALMU. PALMU's advantage on fixed representations therefore does not extend to these MLP settings. These results are consistent with the limitation examined in Section 5.6, where the learned representation also depends on the deleted data. They do not isolate nonconvexity as the cause of the remaining disagreement.

5.4 Stage 2 Verification

The cross-dataset comparisons establish agreement but not why it occurs. We therefore isolate whether generated supervision must be corrected after the forgotten labeled losses are removed. Stage 1 retains the original supervision. Stage 2 updates the labeling model, regenerates supervision, and corrects the downstream classifier. Figure 2 compares the predictions after each stage with those from full retraining.

For this verification, we use a separate CommitBERT/BERT setting where pseudo-labels remain useful after deletion. PALMU outperforms labeled-only retraining in all ten requests at every deletion ratio, with mean macro-F1 gains of 1.56 to 1.83 percentage points. Discarding pseudo labels would remove useful supervision in this setting. We therefore test whether regenerating that supervision after deletion improves agreement with full retraining.

Changes in generated supervision. Updating the labeling classifier on retained data changes hard labels for 1.92%, 2.96%, 4.70%, and 5.64% of the complete unlabeled pool at the four deletion ratios, respectively. These rates are measured over the complete unlabeled pool before confidence filtering. The supervision used for training changes only when an example enters or leaves the accepted set, or receives a different label while remaining accepted. Stage 1 retains the original contributions and cannot reflect either change. These percentages are neither pseudo-label error rates nor evidence of forget-set leakage.

Agreement with full retraining and test performance. Stage 1 discrepancy increases with the deletion ratio, while Stage 2 reduces TV in all 40 paired requests, with reductions of 85.5 to 91.8% in the ratio-specific means. Macro-F1 decreases after Stage 2 in 39 of the 40 requests. Stage 2 improves agreement with full retraining even though test accuracy does not improve. Evaluating accuracy alone would miss this correction.

Separating the two sources of error. To separate downstream optimization error from supervision mismatch, we optimize the downstream classifier to numerical convergence while holding PALMU's finite-step regenerated supervision fixed. Mean TV remains 0.00075, 0.00152, 0.00251, and 0.00332 at the four ratios, respectively. Further downstream optimization alone is therefore insufficient when regenerated supervision still differs from that of the fully retrained labeling classifier. When both classifiers converge on the retained-data pipeline, their predictions agree with scratch retraining to numerical precision.

Together, these results support the two sources of error in Proposition 1. One is remaining downstream optimization error. The other is disagreement between the regenerated supervision and the supervision produced by the fully retrained labeling model. Stage 2 substantially corrects the discrepancy from retaining pre-deletion supervision. These results support regeneration and downstream correction. They do not establish that the preliminary Stage 1 downstream update is necessary. Section 5.5 tests whether that update is needed.

5.5 Is PALMU's Intermediate Update Necessary?

Stage 2 establishes the need to correct generated supervision, but not the need for PALMU's preliminary Stage 1 downstream update. At the matched budget of two downstream updates, starting directly from $\mathbf{w}_0$ and applying both updates after regeneration lowers TV and runtime in all 40 paired cases, with mean macro-F1 within 0.007 percentage points of PALMU at every ratio (Appendix Table 5). The intermediate update therefore provides no measured advantage : regeneration followed by direct downstream correction is sufficient for the observed agreement. The one-update variant is cheaper but fails the individual tolerances twice at 30% deletion, whereas the two-update variant passes all 40. Both variants are faster than warm-started full retraining. Appendix F gives the gradient-descent and limited-memory BFGS (Liu & Nocedal, 1989) budgets and full runtime comparison.

5.6 Architecture Sensitivity

The preceding evidence concerns fixed representations. We next test the boundary where the representation itself is learned from deletion-affected labeled data (Appendix Table 6). All 160 requests on the four deep fixed representations meet both tolerances, with mean TV from 0.00034 to 0.00113 at 10% deletion, and logistic regression passes 159 of 160. In contrast, PALMU misses the single-reference criteria on all 160 MLP requests. The hidden layers also depend on deleted records and remain unchanged by the head correction. Because MLP retraining is nonconvex, we compare each request with three independent retrainings.On CommitBERT and EDM, PALMU's disagreement exceeds the measured variation among retrained models at every deletion ratio. On AG News and CIFAR-10, variation across initializations makes the results less conclusive at lower deletion ratios, but does not explain the disagreement at higher ratios (Appendix G). Adding 1 to 10 epochs of downstream hidden-layer updates after regeneration still does not reach tolerance on either commit dataset at 10% deletion. This diagnostic leaves the labeling model's hidden layers fixed, so it does not test correction of both learned representations. These results support updating the classifier heads when they are the only trainable components affected by deletion. They do not show that the same update is sufficient when the representations also depend on the deleted data.

5.7 Frozen Llama Representations

Finally, we test whether the fixed-representation result extends to a much larger frozen encoder. Table 2 reports the 10% comparison, and Appendix Table 9 extends frozen Llama-3.1-8B features across all four deletion ratios on the two commit datasets. PALMU stays close to full retraining, with 38 of 40 CommitBERT and 40 of 40 EDM requests passing both tolerances. This extends the evidence beyond frozen BERT and ResNet while preserving the requirement that representations remain independent of downstream labeled fitting. Its macro-F1 matches full retraining on CommitBERT while labeled-only retraining is already close on EDM, so the utility of retaining pseudo-labels remains task dependent. These updates act only on classifiers above cached features and do not unlearn the 8B encoder. Appendix H gives the full comparison and timings.

6 Discussion and Limitations

Our experiments show that generated supervision must be considered when defining deletion in self training. Removing labeled records while retaining their generated supervision can leave disagreement with full retraining. On four datasets with fixed representations, PALMU achieves closer predictive agreement than the evaluated baselines using the same reference. Stage 2 improves agreement even when macro F1 decreases, showing why test performance alone cannot establish successful unlearning. The matched budget ablation supports regeneration followed by direct downstream correction and finds no benefit from the preliminary classifier update in that setting. Preserving pseudo labels is especially useful when discarding them reduces predictive utility, as observed in the focused CommitBERT verification study.

Proposition 1 separates remaining downstream optimization error from disagreement in regenerated supervision. The bound applies to regularized linear heads on fixed representations, depends on the retraining reference, and can be loose when regularization is small. It does not certify deletion or provide a privacy guarantee. The MLP results show that the evaluated correction does not adequately address representations learned from deleted data. Extending the method requires correcting both the labeling and downstream representations.

The evaluation uses random deletion requests from fixed data splits, rather than sequential deletion histories. CommitBERT uses weak labels, and the commit datasets lack repository disjoint splits. Baseline comparisons use finite search budgets. Frozen encoder pretraining is outside the deletion request, and retained research checkpoints would require separate handling in deployment. These conditions limit the scope of the findings.

7 Conclusion

Machine unlearning in self training must account for supervision generated from the deleted labeled records. PALMU updates the labeling model, regenerates pseudo labels and acceptance decisions, and corrects the downstream classifier. Across four datasets with fixed representations, it closely approximates full retraining while preserving useful supervision. Stage 2 improves predictive agreement even when test performance decreases, demonstrating why accuracy alone is insufficient to evaluate deletion. The matched budget ablation shows that the preliminary downstream update can be omitted in the tested setting. The learned representation experiments identify the next challenge, correcting the representations themselves alongside the supervision they generate.

AI USE STATEMENT

Generative AI tools were used to assist with drafting, editing, and polishing the text of the manuscript, as well as generating LaTeX code for tables and figures. AI assistance was also occasionally used for light code debugging and minor script formatting. All experimental designs, mathematical formulations, implementation logic, data interpretations, and final manuscript content were independently developed and verified by the authors.

ETHICS STATEMENT

The experiments concern deletion of labeled training records in classification datasets. Predictive agreement and the reported membership attack are limited diagnostics and should not be treated as certification that personal information has been removed. Any deployment would also need to address retained checkpoints, generated targets, and other stored state.

REPRODUCIBILITY STATEMENT

Appendix A gives the proof, Appendix B specifies the update, and Appendices C–E describe the experimental protocols and baseline adaptations. The supplementary tables report all evaluated deletion fractions. The experiment scripts retain configurations, source snapshots, feature manifests, and per-request outputs to support checks against the reported results.

REFERENCES

Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International joint conference on neural networks (IJCNN)*, pp. 1–8. IEEE, 2020.

Larry Armijo. Minimization of functions having lipschitz continuous first partial derivatives. *Pacific Journal of mathematics*, 16(1):1–3, 1966.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE symposium on security and privacy (SP)*, pp. 141–159. IEEE, 2021.

Rishav Chourasia and Neil Shah. Forget unlearning: Towards true data-deletion in machine learning. In *International conference on machine learning*, pp. 6028–6073. PMLR, 2023.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

Chongyu Fan, Jiancheng Liu, Yihua Zhang, Eric Wong, Dennis Wei, and Sijia Liu. Salun: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. *arXiv preprint arXiv:2310.12508*, 2023.

Chongyu Fan, Jiancheng Liu, Licong Lin, Jinghan Jia, Ruiqi Zhang, Song Mei, and Sijia Liu. Simplicity prevails: Rethinking negative preference optimization for llm unlearning. *Advances in Neural Information Processing Systems*, 38:1540–1567, 2026.

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

Kristian G Georgiev, Roy Rinberg, Sam Park, Shivam Garg, Andrew Ilyas, Aleksander Madry, and Seth Neel. Machine unlearning via simulated oracle matching. In *International Conference on Learning Representations*, volume 2025, pp. 49693–49731, 2025.

Niki Gitinabard, Zhikai Gao, Sarah Heckman, Tiffany Barnes, and Collin F Lynch. Analysis of student pair teamwork using github activities. *Journal of Educational Data Mining*, 15(1):32–62, 2023.

Aditya Golatkar, Alessandro Achille, and Stefano 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*, pp. 9304–9312, 2020.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Rushi Jiao, Yichi Zhang, Le Ding, Bingsen Xue, Jicong Zhang, Rong Cai, and Cheng Jin. Learning with limited annotations: a survey on deep semi-supervised learning for medical image segmentation. *Computers in Biology and Medicine*, 169:107840, 2024.

Bjørn Aslak Juliussen, Jon Petter Rui, and Dag Johansen. Algorithms that forget: Machine unlearning and the right to erasure. *Computer Law & Security Review*, 51:105885, 2023.

Tae Hwan Jung. Commitbert: Commit message generation using pre-trained programming language model. In *Proceedings of the 1st Workshop on Natural Language Processing for Programming (NLP4Prog 2021)*, pp. 26–33, 2021.

Georgios Kostopoulos and Sotiris Kotsiantis. Exploiting semi-supervised learning in the education field: A critical survey. *Advances in Machine Learning/Deep Learning-based Technologies: Selected Papers in Honour of Professor Nikolaos G. Bourbakis–Vol. 2*, pp. 79–94, 2021.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

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

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D Li, Ann-Kathrin Dombrowski, Shashwat Goel, Long Phan, et al. The wmdp benchmark: Measuring and reducing malicious use with unlearning. *arXiv preprint arXiv:2403.03218*, 2024.

Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1):503–528, 1989.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C Lipton, and J Zico Kolter. Tofu: A task of fictitious unlearning for llms. *arXiv preprint arXiv:2401.06121*, 2024.

Thanh Tam Nguyen, Thanh Trung Huynh, Zhao Ren, Phi Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen. A survey of machine unlearning. *ACM Transactions on Intelligent Systems and Technology*, 16(5):1–46, 2025.

Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 2006.

Chao Pan, Jin Sima, Saurav Prakash, Vishal Rana, and Olgica Milenkovic. Machine unlearning of federated clusters. *arXiv preprint arXiv:2210.16424*, 2022.

Zhuoyi Peng, Yixuan Tang, and Yi Yang. Adversarial mixup unlearning. In *International Conference on Learning Representations*, volume 2025, pp. 76043–76062, 2025.

Shaofei Shen, Chenhao Zhang, Yawen Zhao, Alina Bialkowski, Weitong Chen, and Miao Xu. Label-agnostic forgetting: A supervision-free unlearning in deep models. In *International Conference on Learning Representations*, volume 2024, pp. 54217–54235, 2024.

Weijia Shi, Jaechan Lee, Yangsibo Huang, Sadhika Malladi, Jieyu Zhao, Ari Holtzman, Daogao Liu, Luke Zettlemoyer, Noah Smith, and Chiyuan Zhang. Muse: Machine unlearning six-way evaluation for language models. In *International Conference on Learning Representations*, volume 2025, pp. 27797–27818, 2025.

Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020.

Zixing Song, Xiangli Yang, Yifei Zhang, Xinyu Fu, Zenglin Xu, and Irwin King. A systematic survey on federated semi-supervised learning. In *IJCAI*, volume 16, pp. 18, 2024.

Yi Tang, Yi Gao, Yonggang Luo, Jucheng Yang, Miao Xu, and Min-Ling Zhang. Unlearning from weakly supervised learning. In *IJCAI*, pp. 5000–5008, 2024.

Yihan Wang, Yiwei Lu, Guojun Zhang, Franziska Boenisch, Adam Dziedzic, Yaoliang Yu, and Xiao-Shan Gao. Muc: Machine unlearning for contrastive learning with black-box evaluation. *arXiv preprint arXiv:2406.03603*, 2024.

Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Advances in neural information processing systems*, 34:18408–18419, 2021.

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning. *arXiv preprint arXiv:2404.05868*, 2024.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

Jianing Zhu, Bo Han, Jiangchao Yao, Jianliang Xu, Gang Niu, and Masashi Sugiyama. Decoupling the class label and the target concept in machine unlearning. In *International Conference on Learning Representations*, volume 2026, pp. 30100–30146, 2026.

A Mathematical Derivations

A.1 Distance to Full Retraining

Assumptions and curvature. The representation is fixed independently of fitting on the deletion affected labeled data. The downstream logits are linear in the classifier parameters $\mathbf{w}$, including the intercept. We assume $\alpha \geq 0$, $n_u > 0$, and $\lambda > 0$, with quadratic regularization applied to every classifier parameter. For fixed labeling parameters $\boldsymbol{\theta}$, the pseudo labels and acceptance decisions do not vary with $\mathbf{w}$.

For one example, let $\mathbf{p}$ be its predicted class probability vector. The cross entropy Hessian with respect to the logits is

$$\mathbf{S} = \operatorname{diag}(\mathbf{p}) - \mathbf{p}\mathbf{p}^\top.$$

For any vector $\mathbf{z}$, with $\mu = \sum_c p_c z_c$, we have

$$\mathbf{z}^\top \mathbf{S}\mathbf{z} = \sum_c p_c z_c^2 - \left(\sum_c p_c z_c\right)^2 = \sum_c p_c (z_c - \mu)^2 \geq 0.$$

Thus $\mathbf{S}$ is positive semidefinite. Since the logits are linear in $\mathbf{w}$, their Jacobian $\mathbf{J}$ is constant in $\mathbf{w}$, and the example contributes $\mathbf{J}^\top \mathbf{S}\mathbf{J}$ to the classifier Hessian. This matrix is also positive semidefinite.

The labeled and accepted pseudo labeled losses have nonnegative weights. Their Hessians therefore sum to a positive semidefinite matrix $\mathbf{H}_{\mathrm{CE}}$. Including regularization gives

$$\nabla^2_{\mathbf{w}} \mathcal{L}_{\mathrm{self}}(\mathbf{w}; \mathcal{D}_r, \boldsymbol{\theta}) = \mathbf{H}_{\mathrm{CE}}(\mathbf{w}; \mathcal{D}_r, \boldsymbol{\theta}) + \lambda \mathbf{I} \succeq \lambda \mathbf{I}.$$

The downstream objective is therefore λ strongly convex and has a unique minimizer for each fixed $\boldsymbol{\theta}$.

Disagreement in generated supervision. Let $\boldsymbol{\theta}_r$ denote PALMU's updated labeling parameters and $\boldsymbol{\theta}_{\mathrm{rt}}$ the labeling parameters obtained by full retraining on the retained labeled data. Define

$$\begin{aligned} \mathcal{C}_{\mathrm{rt}} = \; & \{j \in \{1, \ldots, n_u\} \mid a_j(\boldsymbol{\theta}_r) \neq a_j(\boldsymbol{\theta}_{\mathrm{rt}})\} \\ & \cup \{j \in \{1, \ldots, n_u\} \mid a_j(\boldsymbol{\theta}_r) = a_j(\boldsymbol{\theta}_{\mathrm{rt}}) = 1, \quad \hat{y}_j(\boldsymbol{\theta}_r) \neq \hat{y}_j(\boldsymbol{\theta}_{\mathrm{rt}})\}. \end{aligned}$$

This set contains examples whose active supervision changes. A changed label contributes only when the example is accepted by both labeling models. Define

$$B = \frac{\alpha\sqrt{2}}{n_u} \sum_{j \in \mathcal{C}_{\mathrm{rt}}} \|\mathbf{h}(\mathbf{x}_j)\|_2,$$

where $\mathbf{h}(\mathbf{x}_j)$ includes the constant feature corresponding to the regularized intercept.

Proposition 1. *Under the assumptions above, let $\mathbf{w}_{\mathrm{rt}}$ be the exact minimizer of $\mathcal{L}_{\mathrm{self}}(\mathbf{w}; \mathcal{D}_r, \boldsymbol{\theta}_{\mathrm{rt}})$. For any downstream classifier parameters $\mathbf{w}$,*

$$\|\mathbf{w} - \mathbf{w}_{\mathrm{rt}}\|_F \leq \frac{\|\nabla_{\mathbf{w}} \mathcal{L}_{\mathrm{self}}(\mathbf{w}; \mathcal{D}_r, \boldsymbol{\theta}_r)\|_F + B}{\lambda}. \tag{4}$$

Vectorized classifier parameters use the Euclidean norm, equivalently the Frobenius norm of the corresponding weight matrix.

Proof. Let f be a differentiable λ strongly convex function with minimizer $\mathbf{w}^*$. Strong convexity gives

$$\langle \nabla f(\mathbf{w}) - \nabla f(\mathbf{w}^*), \mathbf{w} - \mathbf{w}^* \rangle \geq \lambda \|\mathbf{w} - \mathbf{w}^*\|_F^2.$$

Since $\nabla f(\mathbf{w}^*) = 0$, the Cauchy–Schwarz inequality gives

$$\lambda \|\mathbf{w} - \mathbf{w}^*\|_F^2 \leq \langle \nabla f(\mathbf{w}), \mathbf{w} - \mathbf{w}^* \rangle \leq \|\nabla f(\mathbf{w})\|_F \|\mathbf{w} - \mathbf{w}^*\|_F.$$

If $\mathbf{w} = \mathbf{w}^*$, the bound holds immediately. Otherwise, dividing by $\|\mathbf{w} - \mathbf{w}^*\|_F$ gives

$$\|\mathbf{w} - \mathbf{w}^*\|_F \leq \frac{\|\nabla f(\mathbf{w})\|_F}{\lambda}. \tag{5}$$

The same argument applies after a classifier weight matrix is vectorized. It also covers the intercept because the intercept is included in the regularization.

For concise notation, define

$$F_r(\mathbf{w}) = \mathcal{L}_{\text{self}}(\mathbf{w}; \mathcal{D}_r, \boldsymbol{\theta}_r), \qquad F_{\text{rt}}(\mathbf{w}) = \mathcal{L}_{\text{self}}(\mathbf{w}; \mathcal{D}_r, \boldsymbol{\theta}_{\text{rt}}).$$

The full retraining head satisfies

$$\mathbf{w}_{\text{rt}} = \arg\min_{\mathbf{w}} F_{\text{rt}}(\mathbf{w}).$$

The retained labeled terms and regularization terms are identical in F_r and F_{rt}. Their gradient difference therefore depends only on the generated supervision.

Consider a pool example accepted under both $\boldsymbol{\theta}_r$ and $\boldsymbol{\theta}_{\text{rt}}$. If its hard label changes, the difference between its two gradient contributions has Frobenius norm

$$\frac{\alpha}{n_u} \left\| \mathbf{h}(\mathbf{x}_j) \left(\mathbf{e}_{\hat{y}_j(\boldsymbol{\theta}_r)} - \mathbf{e}_{\hat{y}_j(\boldsymbol{\theta}_{\text{rt}})} \right)^\top \right\|_F = \frac{\alpha\sqrt{2}}{n_u} \|\mathbf{h}(\mathbf{x}_j)\|_2.$$

If acceptance changes, one objective contains the pseudo label contribution while the other does not. Its norm is bounded by

$$\frac{\alpha}{n_u} \|\mathbf{h}(\mathbf{x}_j)\|_2 \, \|p(\mathbf{x}_j; \mathbf{w}) - \mathbf{e}_y\|_2 \leq \frac{\alpha\sqrt{2}}{n_u} \|\mathbf{h}(\mathbf{x}_j)\|_2.$$

The final inequality follows because $p(\mathbf{x}_j; \mathbf{w})$ is a probability vector and $\|p - \mathbf{e}_y\|_2 \leq \sqrt{2}$.

Unchanged active contributions cancel. Summing over $\mathcal{C}_{\text{rt}}$ and applying the triangle inequality gives

$$\|\nabla_{\mathbf{w}} F_r(\mathbf{w}) - \nabla_{\mathbf{w}} F_{\text{rt}}(\mathbf{w})\|_F \leq B. \tag{6}$$

It follows that

$$\|\nabla_{\mathbf{w}} F_{\text{rt}}(\mathbf{w})\|_F \leq \|\nabla_{\mathbf{w}} F_r(\mathbf{w})\|_F + B.$$

Applying Equation equation 5 to F_{rt} proves

$$\|\mathbf{w} - \mathbf{w}_{\text{rt}}\|_F \leq \frac{\|\nabla_{\mathbf{w}} \mathcal{L}_{\text{self}}(\mathbf{w}; \mathcal{D}_r, \boldsymbol{\theta}_r)\|_F + B}{\lambda},$$

which is Equation equation 4.

Without B, Equation equation 5 bounds the distance from $\mathbf{w}$ to the minimizer of the objective defined by $\boldsymbol{\theta}_r$. The term B accounts for the additional difference between supervision generated by $\boldsymbol{\theta}_r$ and supervision generated by $\boldsymbol{\theta}_{\text{rt}}$. Neither bound describes a distribution of released models or provides a privacy guarantee. Computing B requires the fully retrained labeling model, so this is a reference dependent bound rather than a certificate available from PALMU alone. The bound can be loose when λ is small. It bounds classifier parameter distance and does not directly guarantee a predictive TV tolerance.

Numerical reference error. The mathematical reference $\mathbf{w}_{\text{rt}}$ is the exact minimizer of F_{rt}. In the experiments, the numerical reference $\widetilde{\mathbf{w}}_{\text{rt}}$ is fitted until

$$\|\nabla_{\mathbf{w}} F_{\text{rt}}(\widetilde{\mathbf{w}}_{\text{rt}})\|_F \leq \epsilon_{\text{rt}},$$

with $\epsilon_{\text{rt}} = 10^{-7}$. Strong convexity gives

$$\|\widetilde{\mathbf{w}}_{\text{rt}} - \mathbf{w}_{\text{rt}}\|_F \leq \frac{\epsilon_{\text{rt}}}{\lambda}.$$

The triangle inequality therefore gives

$$\|\mathbf{w} - \widetilde{\mathbf{w}}_{\text{rt}}\|_F \leq \frac{\|\nabla_{\mathbf{w}} \mathcal{L}_{\text{self}}(\mathbf{w}; \mathcal{D}_r, \boldsymbol{\theta}_r)\|_F + B + \epsilon_{\text{rt}}}{\lambda}.$$

This additional term separates numerical reference error from the two error sources in Proposition 1.

B ALGORITHM AND NUMERICAL IMPLEMENTATION

Algorithm 1 PALMU for one deletion request

Require: classifiers $\boldsymbol{\theta}_0, \mathbf{w}_0$; sets $\mathcal{D}_\ell, \mathcal{D}_f, \mathcal{D}_u$; stored $\hat{y}_j(\boldsymbol{\theta}_0), a_j(\boldsymbol{\theta}_0)$; selected $\alpha, \lambda, \lambda_\ell, \tau$; budgets K_ℓ, K_1, K_2
Ensure: updated downstream classifier $\mathbf{w}_2$

$\mathcal{D}_r \leftarrow \mathcal{D}_\ell \setminus \mathcal{D}_f$
// Stage 1 optimizes the retained objective with stored supervision
$\mathbf{w} \leftarrow \mathbf{w}_0$
for $t = 1, \ldots, K_1$ **do**
  $\mathbf{g} \leftarrow \nabla_{\mathbf{w}} \mathcal{L}_{\mathrm{self}}(\mathbf{w}; \mathcal{D}_r, \boldsymbol{\theta}_0)$
  solve $\mathbf{H}_{\mathrm{self}}(\mathbf{w}; \mathcal{D}_r, \boldsymbol{\theta}_0)\mathbf{v} = \mathbf{g}$ using conjugate gradients with scalar preconditioning
  choose η using Armijo backtracking and set $\mathbf{w} \leftarrow \mathbf{w} - \eta\mathbf{v}$
end for
$\mathbf{w}_1 \leftarrow \mathbf{w}$
// Stage 2 updates the labeling classifier
$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}_0$
for $t = 1, \ldots, K_\ell$ **do**
  $\mathbf{g}_\ell \leftarrow \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathrm{lab}}(\boldsymbol{\theta}; \mathcal{D}_r)$
  solve $\mathbf{H}_{\mathrm{lab}}(\boldsymbol{\theta}; \mathcal{D}_r)\mathbf{v}_\ell = \mathbf{g}_\ell$ using conjugate gradients with scalar preconditioning
  choose η_ℓ using Armijo backtracking and set $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta_\ell \mathbf{v}_\ell$
end for
$\boldsymbol{\theta}_r \leftarrow \boldsymbol{\theta}$
Regenerate $\hat{y}_j(\boldsymbol{\theta}) = \arg\max_c p_c(\mathbf{x}_j; \boldsymbol{\theta})$ and $a_j(\boldsymbol{\theta}) = \mathbf{1}\{\max_c p_c(\mathbf{x}_j; \boldsymbol{\theta}) \geq \tau\}$ for every $\mathbf{x}_j \in \mathcal{D}_u$.
$\mathcal{C} \leftarrow \{j \mid a_j(\boldsymbol{\theta}_r) \neq a_j(\boldsymbol{\theta}_0)$ or $[a_j(\boldsymbol{\theta}_r) = a_j(\boldsymbol{\theta}_0) = 1$ and $\hat{y}_j(\boldsymbol{\theta}_r) \neq \hat{y}_j(\boldsymbol{\theta}_0)]\}$
$\mathbf{w}_2 \leftarrow \mathbf{w}_1$
if $\alpha > 0$ **and** $\mathcal{C} \neq \varnothing$ **then**
  for $t = 1, \ldots, K_2$ **do**
    $\mathbf{g} \leftarrow \nabla_{\mathbf{w}} \mathcal{L}_{\mathrm{self}}(\mathbf{w}_2; \mathcal{D}_r, \boldsymbol{\theta}_r)$
    solve $\mathbf{H}_{\mathrm{self}}(\mathbf{w}_2; \mathcal{D}_r, \boldsymbol{\theta}_r)\mathbf{v} = \mathbf{g}$ using conjugate gradients with scalar preconditioning
    choose η using Armijo backtracking and set $\mathbf{w}_2 \leftarrow \mathbf{w}_2 - \eta\mathbf{v}$
  end for
end if
return $\mathbf{w}_2$

Stage 1 selects the budget K_1 on development data and sets the step size during each update with the standard Armijo backtracking rule (Nocedal & Wright, 2006). The search starts with $\eta_1^{(t)} = 1$ and repeatedly halves it until

$$\mathcal{L}_{\mathrm{self}}\left(\mathbf{w}_1^{(t)} - \eta_1^{(t)}\mathbf{v}_1^{(t)}; \mathcal{D}_r, \boldsymbol{\theta}_0\right) \leq \mathcal{L}_{\mathrm{self}}(\mathbf{w}_1^{(t)}; \mathcal{D}_r, \boldsymbol{\theta}_0) - 10^{-4}\eta_1^{(t)}\left\langle \mathbf{g}_1^{(t)}, \mathbf{v}_1^{(t)} \right\rangle. \tag{7}$$

The first step size satisfying this condition is used, and the implementation tests at most 35 values. No validation data enter this search. The labeled loss remains normalized by $|\mathcal{D}_r|$, while the pseudo-label loss keeps the raw pool normalization n_u of the self-training objective in Section 3.

For fixed representations, both $\mathcal{L}_{\mathrm{lab}}$ and $\mathcal{L}_{\mathrm{self}}$ are regularized multiclass softmax objectives. Let $\mathcal{J}$ denote either objective with its generated supervision held fixed. Let $\mathbf{X}$ contain the representations used by $\mathcal{J}$, let $\boldsymbol{\omega}$ contain their loss weights, and let $\mathbf{P}$ contain the predicted class probabilities. The weights in $\boldsymbol{\omega}$ are $1/|\mathcal{S}|$ for labeled examples and α/n_u for accepted pool examples. Let $\lambda_{\mathcal{J}}$ be λ_ℓ for the labeling objective and λ for the downstream objective.

For a matrix $\mathbf{V}$ with the same shape as the classifier parameters, the Hessian product is

$$\mathbf{H}_{\mathcal{J}}[\mathbf{V}] = \mathbf{X}^\top\left[(\boldsymbol{\omega}\mathbf{1}_C^\top) \odot \mathbf{P} \odot \left(\mathbf{X}\mathbf{V} - \left((\mathbf{P} \odot \mathbf{X}\mathbf{V})\mathbf{1}_C\right)\mathbf{1}_C^\top\right)\right] + \lambda_{\mathcal{J}}\mathbf{V}. \tag{8}$$

Here $\mathbf{1}_C$ is the length C vector of ones and $\odot$ denotes elementwise multiplication. The shared softmax denominator produces the terms connecting different classes.

Each Newton direction is obtained by solving its Hessian system with conjugate gradients and Hessian-vector products, so the solver never forms the Hessian or its inverse. It uses the mean Hessian diagonal as a scalar preconditioner. Positive regularization supplies the curvature term in Equation equation 8, so the reported runs use no additional Newton damping. Each update recomputes the gradient and the Hessian product and selects its step size with the Armijo rule in Equation equation 7, and the reported updates use no step norm cap.

For the algorithm, write $\mathbf{H}_{\mathrm{lab}} = \nabla^2_{\boldsymbol{\theta}} \mathcal{L}_{\mathrm{lab}}$ and $\mathbf{H}_{\mathrm{self}} = \nabla^2_{\mathbf{w}} \mathcal{L}_{\mathrm{self}}$.

The four-dataset comparison uses a common selected budget $K_\ell = K_1 = K_2 = K$ within each setting. The selected values are reported in Table 3. The initial confirmation uses $K_\ell = K_1 = K_2 = 1$.

The variant without the intermediate downstream update omits the Stage 1 block. It updates the labeling classifier, regenerates supervision, and then optimizes $\mathcal{L}_{\mathrm{self}}(\mathbf{w}; \mathcal{D}_r, \boldsymbol{\theta}_r)$ directly from $\mathbf{w}_0$ for its declared downstream budget. This objective still excludes the forgotten labeled examples. If all pseudo-labels and acceptance decisions remain unchanged, the stored and regenerated downstream objectives are identical. This does not imply that a finite Stage 1 budget has reached their common minimizer.

All head optimization uses double precision. The conjugate-gradient solver uses a relative residual tolerance of 10^{-8}, checks the true residual every 500 iterations, and permits at most 2,000 iterations. The Armijo search starts at one, uses coefficient 10^{-4}, and tests at most 35 candidate step sizes by repeated halving. Numerical reference heads must reach gradient norm 10^{-7} within 80 Newton updates.

For an MLP, the declared training procedure first applies its selected optimizer to the hidden layers and then fits the conditional head to the stated gradient tolerance. Convergence of the conditional head does not establish convergence or uniqueness of the learned hidden representation.

C Datasets and Representations

CommitBERT. We use commit messages from the Python portion of CommitBERT (Jung, 2021). A message receives the Test label if it contains `test`, `spec`, `unittest`, `pytest`, or `assert`. Otherwise, `commit_type=fix` gives Bug-fix and the remaining messages receive Improvement. These are weak labels, not independently annotated semantic classes. Each request seed samples 333 labeled examples per class. The raw unlabeled pool contains 5,000 records. Exact normalized-text matches across training and test are removed, as are conflicting duplicates within a source. These checks do not provide a repository-disjoint split.

EDM. The labeled data retain the supplied Improvement, Bug-fix, and Test tags and exclude other tags. Cleaning leaves 946 labeled records, split into 477 training, 140 validation, 140 audit, and 189 test examples. The unlabeled pool contains 5,000 disjoint records. The experiment uses the text and supplied tags rather than numeric features in the input CSV files. The data originate from educational software-development activity (Gitinabard et al., 2023). The available tags are not described here as independently reannotated ground truth.

AG News and CIFAR-10. AG News uses its four native classes, with 200 labeled records per class, 10,000 pool records, 400 validation records, 400 audit records, and the official 7,600-record test set. CIFAR-10 uses 200 labeled images per class, a 10,000-image pool, 500 validation images, 500 audit images, and the official 10,000-image test set. These subsets are fixed before request sampling. The class counts differ from the commit datasets and are retained by the multiclass implementation.

Features. For the text LR and MLP experiments, one TF-IDF map is fitted only on the disjoint unlabeled pool and produces 300 features. CIFAR-10 uses a 300-dimensional PCA map fitted on the pool. The shared input maps are fixed before labeled fitting. The MLP then learns ReLU hidden layers of widths 128 and 64. It is this learned representation, not the input map, that violates the fixed-feature assumption for a head-only deletion update.

Table 3: Selected parameters for the four-dataset and Llama comparisons. K is the Newton budget for each PALMU component. A zero pseudo-label weight deactivates the Stage 2 classifier correction.

Dataset	Model	λ_ℓ	λ	α	τ	K
CommitBERT	LR	0.01	0.01	0	0.7	1
CommitBERT	MLP	0.01	0.01	0.25	0.5	3
CommitBERT	BERT	5e-05	5e-05	1	0.5	1
EDM	LR	0.001	0.0001	0.25	0.5	1
EDM	MLP	0.01	0.01	1	0.7	1
EDM	BERT	3e-05	3e-05	0.25	0.9	1
AG News	LR	0.001	0.001	0.25	0.7	1
AG News	MLP	0.001	0.0001	1	0.7	1
AG News	BERT	0.0001	0.0001	1	0.35	1
CIFAR-10	LR	0.0001	0.0001	1	0.7	1
CIFAR-10	MLP	0.001	0.001	0	0.35	1
CIFAR-10	ResNet-50	0.0001	0.0001	0.25	0.35	1
CommitBERT	Llama	1e-05	3e-06	1	0.7	1
EDM	Llama	0.0001	0.0001	0.25	0.9	1

Frozen BERT features use `bert-base-uncased` (Devlin et al., 2019), CLS pooling, maximum sequence length 64, and per-record normalization. The image encoder is an ImageNet-pretrained ResNet-50. Llama features use Llama-3.1-8B revision `d04e592b`, mean pooling, maximum length 64, and 4-bit loading during extraction. All these encoders are fixed before downstream fitting. Their pretraining data are outside the deletion request. The head includes a regularized intercept.

D Development and Evaluation Protocols

Development uses validation examples and seeds 0–2 at 10% deletion. The labeling classifier is selected before the self-trained classifier. Candidate self-training configurations include $\alpha = 0$, and the confidence threshold applies to the maximum predicted class probability. The grids and their selected values are setting specific. In particular, the AG News and CIFAR-10 protocols include confidence threshold 0.35, whereas the initial CommitBERT confirmation uses 0.5, 0.7, and 0.9 for nonzero pseudo-label weight. These choices should not be conflated with an independent sensitivity analysis.

The initial CommitBERT/BERT confirmation uses labeled budget 999 and pool size 5,000. Its regularization grid is $\{3 \times 10^{-6}, 10^{-5}, 3 \times 10^{-5}, 5 \times 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$ and pseudo-label weights are $\{0, 0.25, 1\}$. It selects $\lambda_\ell = 3 \times 10^{-5}$, $\lambda = 5 \times 10^{-5}$, $\alpha = 1$, and $\tau = 0.5$. One labeling update and one Stage 2 classifier update follow one Stage 1 update. Its validation set contains 11,918 additional labeled records, which must be counted when describing label efficiency.

Evaluation phases are kept separate. The confirmation uses seeds 100–109. The optimizer follow-up uses seeds 200–209. The four-dataset and Llama baseline comparisons use seeds 500–509 and rerun PALMU and reference models on their corresponding requests. They are follow-up evaluations on datasets already examined during the project, not independent pristine-test replications. Every deletion request starts from the original model rather than the output of a previous request.

Metrics and uncertainty. Macro-F1 and TV use the same test records for all methods within a request. AUC uses log true-class probability as the attack score and compares forgotten examples with class-matched audit examples sampled without replacement. Audit size can be smaller than the forget set, particularly on EDM. FS is computed per request as $1 - 2|\mathrm{AUC} - 0.5|$ and then averaged. It is not computed by transforming the mean AUC. Tables report means and sample standard deviations across ten request seeds. These deviations describe variation across requests on a fixed split, not uncertainty over independently sampled datasets.

Compute and saved artifacts. The commit LR/MLP runs use a Windows CPU environment with Python 3.11 and PyTorch 2.0.1. The BERT comparison runs use Linux, Python 3.12, and PyTorch 2.7.1. Frozen Llama features are extracted on the server. Head training and deletion run in double

precision with four CPU threads. Timing excludes feature extraction, evaluation, and checkpoint serialization. The optimizer and published-baseline comparisons use three randomized-order repetitions per condition and take the median. Feature manifests, selected configurations, and source snapshots are retained with the experiment outputs.

E Published Baseline Adaptations

The comparison uses classification adaptations of the cited methods. It is not a reproduction of their original image-benchmark scores or an exhaustive comparison with every unlearning algorithm. The implementation audit pins the official code and compares selected reference functions with the adapted implementation. All methods share the data and full retraining reference within an unsharded setting. On MLP, Fisher, SCRUB, and SalUn act on all learned parameters, whereas PALMU acts on the final layers.

SalUn. The implementation ranks absolute forget-loss gradients globally and updates the selected fraction of weights (Fan et al., 2023). Random targets are sampled uniformly over all classes, including the original class. Each epoch performs random-label fitting followed by retained-data fitting. Momentum is 0.9, and masked weights are restored so weight decay and momentum do not move them. Development searches fractions $\{0.1, 0.5, 0.9\}$, learning rates $\{0.001, 0.01, 0.1\}$, and epochs $\{10, 30\}$.

SCRUB. We use SCRUB without rewind (Kurmanji et al., 2023). The original classifier is fixed. The KL direction is from its predictions to the updated classifier, with temperature 2 and the squared-temperature factor. The first three epochs include forget-KL maximization, followed by retained-data minimization. Subsequent epochs minimize retained cross-entropy plus 0.5 times retained KL. The adaptation uses full-batch Adam updates. Development searches learning rates $\{0.0001, 0.0005, 0.001\}$ and epochs $\{10, 30\}$. These results should not be attributed to SCRUB+R.

Fisher. The Fisher adaptation uses expected model-Fisher diagonal noise (Golatkar et al., 2020), not a diagonal Newton deletion step. It averages class-conditioned squared gradients weighted by model probabilities. Fixed linear heads use the analytic diagonal, while MLP uses per-example class gradients. The perturbation retains the audited variance scaling, floor, and clipping. Noise scales are selected from $\{10^{-8}, 10^{-7}, 10^{-6}, 10^{-5}\}$.

Regeneration variants. Each of these three methods has a separately tuned variant that refits the labeling classifier on retained data, regenerates supervision, and then applies the method. This work is recomputed and timed rather than borrowed from the evaluation reference. The wrapper is part of our comparison design. Plain variants retain stored pseudo-labels.

SISA. Our self-training adaptation uses two fixed shards and two cumulative slices per shard (Bourtoule et al., 2021). Each shard has its own unlabeled partition and labeling classifier. Predictions are averaged across shards. A deletion rolls back an affected shard to the checkpoint before its earliest affected slice, then replays the declared training schedule. Assignments keep their original slots. The reference retrains the same sharded procedure on retained data. Consequently zero SISA TV establishes agreement with that reference, not with the unsharded classifier used by other rows. Timings reuse in-memory checkpoints and exclude serialization.

Selection and limits. Each baseline variant is selected separately using validation TV and macro-F1 drop. The stored-supervision and regeneration arms each evaluate 18 SalUn, 6 SCRUB, and 4 Fisher configurations over the development seeds. These candidate counts describe the evaluated grids but do not imply that all methods received equal optimization compute. Admissible candidates satisfy both mean tolerances. Among them, selection minimizes TV and then F1 drop. If no candidate passes, the lowest-TV candidate is retained with its failed margin visible. A candidate that diverges on any development seed is excluded. Some SCRUB and SalUn selections lie on the evaluated grid boundaries. The results do not establish performance under larger training or search budgets. Full-batch updates and classifier-specific regularization are adaptations to this experiment, not claims of equivalence to the original training schedules.

F Stage 2 and Optimizer Results

The confirmation study and optimizer follow-up use different request seeds and remain separate from the baseline comparison. Table 4 includes the confirmation controls. Table 5 shows that, at the same budget of two downstream updates, direct correction after regeneration has lower mean TV and runtime than PALMU at every deletion ratio. Across the 40 paired requests, direct correction lowers both TV and runtime, with mean macro F1 differing from PALMU by at most 0.007 percentage points at each ratio. The direct two update variant passes both individual tolerances in all 40 requests. The direct one update variant fails twice at 30% deletion, despite its mean TV of 0.00794 remaining below 0.01. An admissible mean therefore does not guarantee that every request passes. These results support omitting the intermediate Stage 1 update in this setting while retaining sufficient downstream correction.

Table 4: Initial comparison: mean ± sample standard deviation over ten seeds; time in seconds.

Ratio	Method	Macro-F1	TV	Time
5%	Discard changed pseudo-labels	0.85461 ± 0.00468	0.00611 ± 0.00107	1.92114 ± 0.02227
5%	Exact regeneration + Newton correction	0.85342 ± 0.00506	0.00030 ± 0.00010	0.84165 ± 0.01863
5%	Retraining with original acceptance	0.85403 ± 0.00487	0.00644 ± 0.00073	2.11835 ± 0.03129
5%	Full self-training retraining	0.85344 ± 0.00506	0.00000 ± 0.00000	2.12119 ± 0.02290
5%	Labeled-only retraining	0.83668 ± 0.00625	0.06049 ± 0.00393	0.17164 ± 0.00178
5%	Original model	0.85613 ± 0.00462	0.01709 ± 0.00285	0.00000 ± 0.00000
5%	PALMU	0.85331 ± 0.00492	0.00087 ± 0.00037	0.69928 ± 0.01725
5%	Regeneration + gradient descent	0.85455 ± 0.00458	0.01025 ± 0.00165	0.74067 ± 0.01172
5%	Regeneration + converged classifier	0.85332 ± 0.00498	0.00075 ± 0.00042	1.97518 ± 0.02541
5%	Retain-only gradient descent	0.83922 ± 0.00621	0.03566 ± 0.00188	0.03430 ± 0.00014
5%	Stage 1 only	0.85528 ± 0.00456	0.01058 ± 0.00174	0.34002 ± 0.01034
5%	Retraining with stored pseudo-labels	0.85532 ± 0.00459	0.01059 ± 0.00174	1.94370 ± 0.01417
10%	Discard changed pseudo-labels	0.85342 ± 0.00522	0.00832 ± 0.00123	1.90412 ± 0.06780
10%	Exact regeneration + Newton correction	0.85040 ± 0.00568	0.00068 ± 0.00020	0.83063 ± 0.03005
10%	Retraining with original acceptance	0.85219 ± 0.00534	0.00745 ± 0.00063	2.10617 ± 0.06019
10%	Full self-training retraining	0.85058 ± 0.00586	0.00000 ± 0.00000	2.09525 ± 0.02137
10%	Labeled-only retraining	0.83502 ± 0.00647	0.06035 ± 0.00370	0.16302 ± 0.00499
10%	Original model	0.85613 ± 0.00462	0.02623 ± 0.00437	0.00000 ± 0.00000
10%	PALMU	0.85061 ± 0.00567	0.00170 ± 0.00043	0.69774 ± 0.03435
10%	Regeneration + gradient descent	0.85413 ± 0.00490	0.01558 ± 0.00253	0.73410 ± 0.02545
10%	Regeneration + converged classifier	0.85058 ± 0.00567	0.00152 ± 0.00041	1.96724 ± 0.05404
10%	Retain-only gradient descent	0.83960 ± 0.00593	0.04067 ± 0.00365	0.03253 ± 0.00101
10%	Stage 1 only	0.85476 ± 0.00523	0.01592 ± 0.00251	0.33452 ± 0.01215
10%	Retraining with stored pseudo-labels	0.85485 ± 0.00532	0.01586 ± 0.00246	1.93766 ± 0.04733
20%	Discard changed pseudo-labels	0.85215 ± 0.00528	0.01280 ± 0.00219	1.78964 ± 0.05347
20%	Exact regeneration + Newton correction	0.84663 ± 0.00749	0.00153 ± 0.00042	0.79332 ± 0.02501
20%	Retraining with original acceptance	0.84835 ± 0.00698	0.00843 ± 0.00054	2.03211 ± 0.02684
20%	Full self-training retraining	0.84687 ± 0.00759	0.00000 ± 0.00000	2.04301 ± 0.05403
20%	Labeled-only retraining	0.82876 ± 0.00761	0.06089 ± 0.00382	0.14823 ± 0.00287
20%	Original model	0.85613 ± 0.00462	0.04036 ± 0.00512	0.00000 ± 0.00000
20%	PALMU	0.84703 ± 0.00750	0.00309 ± 0.00088	0.67298 ± 0.02535
20%	Regeneration + gradient descent	0.85308 ± 0.00541	0.02365 ± 0.00372	0.71413 ± 0.02054
20%	Regeneration + converged classifier	0.84711 ± 0.00747	0.00251 ± 0.00080	1.90769 ± 0.04786
20%	Retain-only gradient descent	0.83863 ± 0.00661	0.04879 ± 0.00419	0.03009 ± 0.00078
20%	Stage 1 only	0.85479 ± 0.00434	0.02458 ± 0.00407	0.32740 ± 0.01341
20%	Retraining with stored pseudo-labels	0.85477 ± 0.00428	0.02455 ± 0.00419	1.90119 ± 0.06557
30%	Discard changed pseudo-labels	0.84892 ± 0.00680	0.01384 ± 0.00178	1.69770 ± 0.01722
30%	Exact regeneration + Newton correction	0.84247 ± 0.00914	0.00199 ± 0.00039	0.75005 ± 0.01694
30%	Retraining with original acceptance	0.84446 ± 0.00823	0.00856 ± 0.00081	1.96002 ± 0.02249
30%	Full self-training retraining	0.84270 ± 0.00935	0.00000 ± 0.00000	1.95172 ± 0.02453
30%	Labeled-only retraining	0.82673 ± 0.00836	0.05857 ± 0.00324	0.13175 ± 0.00161
30%	Original model	0.85613 ± 0.00462	0.04841 ± 0.00473	0.00000 ± 0.00000
30%	PALMU	0.84246 ± 0.00919	0.00407 ± 0.00064	0.64533 ± 0.01623
30%	Regeneration + gradient descent	0.85101 ± 0.00686	0.02710 ± 0.00282	0.68666 ± 0.01180
30%	Regeneration + converged classifier	0.84279 ± 0.00909	0.00332 ± 0.00069	1.83773 ± 0.02107
30%	Retain-only gradient descent	0.84077 ± 0.00751	0.05585 ± 0.00496	0.02652 ± 0.00011
30%	Stage 1 only	0.85224 ± 0.00553	0.02810 ± 0.00312	0.31460 ± 0.01106
30%	Retraining with stored pseudo-labels	0.85224 ± 0.00576	0.02788 ± 0.00306	1.82597 ± 0.02034

Reading Table 5 across ratios, PALMU and the two-update variant that omits the intermediate step reach almost the same TV, and both stay far below the 0.01 tolerance at every ratio. The one-update variant is the fastest correction, but its TV grows with the deletion ratio and reaches 0.00794 at 30%, which is why it fails the individual tolerance twice at that ratio even though its mean stays admissible. Full retraining has zero TV by construction and costs roughly two to three times the PALMU time. The pattern shows that the fidelity comes from regenerating the supervision and converging the downstream correction, not from the separate Stage 1 pass.

Table 5: CommitBERT/BERT optimizer comparison, means over seeds 200–209. PALMU uses one Stage 1 and one Stage 2 classifier update. All methods include the corresponding labeling-model work. TV is measured against scratch full self-training retraining for the same request. Times are seconds, averaged over per-request medians of three repetitions.

	5%		10%		20%		30%	
Method	TV	s	TV	s	TV	s	TV	s
PALMU	.00108	.719	.00139	.711	.00269	.687	.00447	.674
No intermediate update, 1 step	.00134	.374	.00200	.371	.00438	.357	.00794	.345
No intermediate update, 2 steps	.00102	.676	.00125	.663	.00232	.642	.00368	.624
Full retraining, initialized	.00000	1.034	.00000	1.011	.00000	1.005	.00000	1.202
Full retraining, scratch	.00000	2.138	.00000	2.111	.00000	2.039	.00000	1.965

Table 6: PALMU fidelity across architectures. TV is the mean at 10% deletion. Passes count requests with TV at most 0.01 and a macro-F1 drop from scratch full self-training retraining of at most 0.01 over ten seeds and four deletion ratios. Each request uses its own scratch-retraining reference. Deep fixed features are BERT for text and ResNet-50 for images.

	LR		MLP		Deep fixed features	
Dataset	TV at 10%	Passes	TV at 10%	Passes	TV at 10%	Passes
CommitBERT	0.00003	40/40	0.03251	0/40	0.00113	40/40
EDM	0.00101	39/40	0.03219	0/40	0.00034	40/40
AG News	0.00017	40/40	0.05845	0/40	0.00107	40/40
CIFAR-10	0.00123	40/40	0.25505	0/40	0.00034	40/40

G MLP Update-Scope Diagnostic

The diagnostic reuses the CommitBERT and EDM MLP evaluation requests. It compares the last-layer PALMU update with 1, 3, 5, and 10 epochs of final-model fine-tuning followed by a converged conditional head fit. Regenerated supervision still comes from a labeling classifier whose hidden layers are fixed. Thus this diagnostic does not test complete retraining of both representations.

Table 6 summarizes the boundary. Over the requests in each architecture, the four deep fixed representations pass every request and logistic regression passes all but one, while no MLP request passes. The rest of this section explains why the MLP result is a genuine limitation rather than an artifact of the reference.

Retraining variability control. The original pass criterion compares each update with one scratch-retraining reference. To measure sensitivity to that reference, we hold each request fixed and run full self-training retraining three times from different initializations. We report the mean pairwise TV among the three references and the range of their test macro-F1 values. "Inside" means that PALMU's TV to the original reference is no larger than the maximum pairwise TV among those three retrainings on the same request. This is an empirical initialization-sensitivity control, not a new tolerance or a claim that PALMU passes.

Table 8: MLP update-scope diagnostic at 10% deletion, means over seeds 500–509. Hidden-update budgets were examined diagnostically and were not selected as a new primary method. Runtime is omitted because the hidden-update arm excludes the preceding target-regeneration work.

	CommitBERT			EDM		
Correction	F1	TV	FS	F1	TV	FS
Full retraining, scratch	0.8655	0.00000	0.9485	0.8130	0.00000	0.8731
Full retraining, initialized	0.8684	0.00809	0.9468	0.8115	0.01151	0.8707
Labeled-only retrain	0.8586	0.04485	0.9488	0.8044	0.05093	0.8779
PALMU	0.8706	0.03251	0.8883	0.8245	0.03219	0.7287
PALMU + hidden, 1 epoch	0.8517	0.05563	0.9321	0.7861	0.06389	0.8652
PALMU + hidden, 3 epochs	0.8591	0.04965	0.9392	0.8038	0.04824	0.8421
PALMU + hidden, 5 epochs	0.8725	0.04252	0.9474	0.8145	0.03763	0.8430
PALMU + hidden, 10 epochs	0.8649	0.03992	0.9253	0.8252	0.04581	0.8173

Table 7: Retrain-to-retrain variability at 10% deletion, averaged over ten fixed requests. Each request has three scratch retrainings. F1 range is the within-request range across those retrainings. The fixed controls use CommitBERT/LR, AG News/BERT, and CIFAR-10/ResNet-50.

Dataset	Retrain–retrain TV	Retrain F1 range	PALMU TV	Inside
MLP (trained hidden)				
CommitBERT	0.006	0.002	0.033	0/10
EDM	0.009	0.005	0.032	0/10
AG News	0.068	0.004	0.058	9/10
CIFAR-10	0.309	0.007	0.255	10/10
Fixed controls				
CommitBERT, LR	0.000	0.000	0.00003	–
AG News, BERT	0.000	0.000	0.00107	–
CIFAR-10, ResNet-50	0.000	0.000	0.00034	–

On CommitBERT and EDM, PALMU lies outside the measured retraining range for all ten requests at every deletion ratio, so their MLP gaps remain last-layer failures. On AG News, the inside counts are 10/10, 9/10, 0/10, and 0/10 from 5% to 30% deletion. On CIFAR-10 they are 10/10, 10/10, 7/10, and 0/10. The control therefore qualifies the low-ratio AG News and CIFAR-10 pass interpretation, but it does not rescue the higher-ratio results. In the three fixed-representation controls, retrain-to-retrain TV is numerically zero at every ratio while PALMU remains close to that unique head solution.

None of the tested hidden-update budgets reaches mean TV 0.01 on either dataset at 10% deletion. Even full training initialized from the original model differs from scratch training because the hidden-layer objective is nonconvex and the training schedule is finite. The results identify a limitation of these updates, not an impossibility theorem for MLP unlearning.

H Complete Method Comparisons

The following tables retain all four deletion fractions for the four datasets and the two Llama settings. Each row reports a mean and sample standard deviation over seeds 500–509. F1 and FS are higher-is-better diagnostics, TV is lower-is-better predictive discrepancy, and time is in seconds. AUC is the raw loss-attack result. SISA uses its own sharded reference wherever marked with a dagger. Figure 4 summarizes the trend across deletion ratios, while the tables report the complete numerical results.

Table 9 reports the comparison at 10% deletion. Across all four deletion ratios, PALMU meets both tolerances in 38 of 40 CommitBERT requests and all 40 EDM requests. At 10% deletion, its macro F1 is close to full retraining on CommitBERT, and its forgetting score differs by less than 0.001. On EDM, labeled-only retraining is already close to the reference, showing that the utility of preserving

Table 9: Frozen Llama-3.1-8B features at 10% deletion, means over seeds 500–509. All updates act on the classifiers above cached representations. FS is the loss-attack forgetting score. †SISA uses its own sharded retraining reference.

	CommitBERT			EDM		
Method	F1↑	TV↓	FS↑	F1↑	TV↓	FS↑
Full self-training retraining	0.9007	0.00000	0.9299	0.8280	0.00000	0.9187
Labeled-only retraining	0.8881	0.06294	0.9413	0.8296	0.00602	0.9186
Fisher + regeneration	0.9037	0.01870	0.9027	0.8278	0.02842	0.8301
SCRUB + regeneration	0.9035	0.01781	0.9050	0.8286	0.02476	0.8471
SalUn + regeneration	0.9036	0.01870	0.9024	0.8285	0.02826	0.8315
SISA†	0.8926	0.00000	0.9349	0.8093	0.00000	0.9244
PALMU	0.9010	0.00294	0.9292	0.8280	0.00017	0.9186

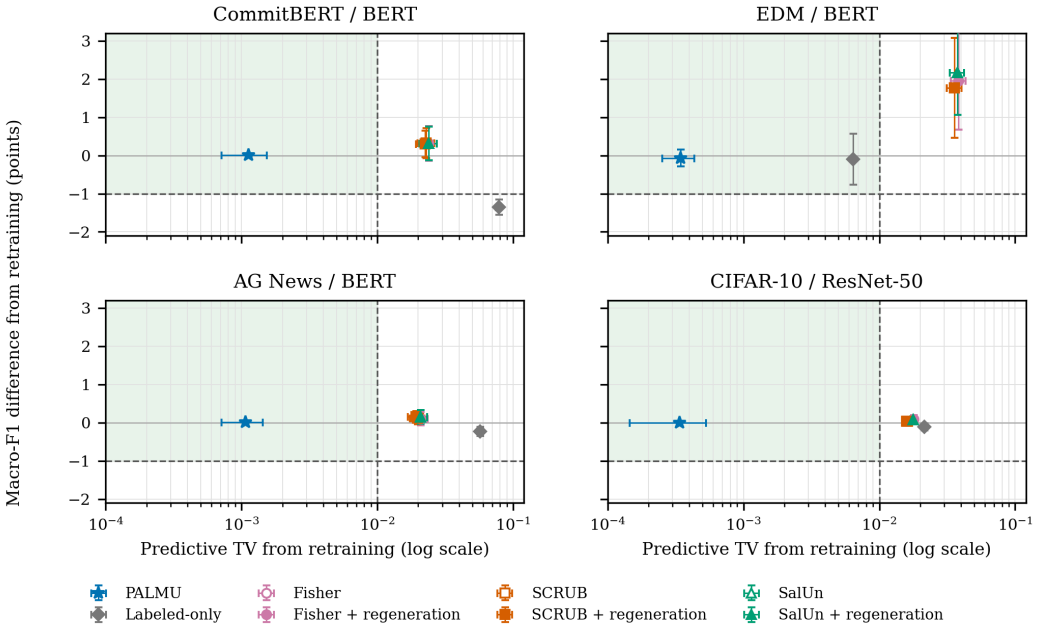

Figure 3: Fidelity and utility at 10% deletion using deep fixed representations. Each point shows the mean across seeds 500 to 509, while the horizontal and vertical bars show sample standard deviations across paired requests. The shaded region includes methods with a mean TV of at most 0.01 and a mean macro-F1 drop of at most one percentage point. Open markers represent stored supervision variants, and filled markers represent regeneration variants. SISA is omitted because it is evaluated against a different sharded reference. Exact values are reported in Table 1.

pseudo labels remains task dependent. All updates act on classifiers above cached representations. The reported update times exclude feature extraction and do not measure unlearning of the frozen encoder.

Figure 3 shows the joint fidelity–utility view of the 10% deep-feature comparison in main Table 1. The error bars describe variation over paired deletion requests on the fixed splits.

Table 10: CommitBERT, BERT. Mean ± sample standard deviation over ten seeds. †SISA uses its own sharded reference.

Forget	Method	F1	TV	FS	AUC	Time (s)
5%	Full retraining	0.8506 ± 0.0072	0.00000 ± 0.00000	0.8986 ± 0.0609	0.4756 ± 0.0558	2.12 ± 0.03
5%	Labeled-only retraining	0.8354 ± 0.0076	0.07842 ± 0.00334	0.9000 ± 0.0617	0.4751 ± 0.0551	0.15 ± 0.01
5%	Fisher	0.8506 ± 0.0078	0.01794 ± 0.00178	0.9118 ± 0.0764	0.5182 ± 0.0570	0.02 ± 0.00
5%	Fisher + regeneration	0.8506 ± 0.0078	0.01794 ± 0.00178	0.9118 ± 0.0764	0.5182 ± 0.0570	0.17 ± 0.01
5%	SCRUB	0.8507 ± 0.0079	0.01717 ± 0.00177	0.9128 ± 0.0765	0.5163 ± 0.0573	0.14 ± 0.00
5%	SCRUB + regeneration	0.8510 ± 0.0078	0.01660 ± 0.00171	0.9125 ± 0.0763	0.5166 ± 0.0572	0.30 ± 0.01

Forget	Method	F1	TV	FS	AUC	Time (s)
5%	SalUn	0.8507 ± 0.0078	0.01797 ± 0.00176	0.9123 ± 0.0764	0.5181 ± 0.0569	0.05 ± 0.00
5%	SalUn + regeneration	0.8506 ± 0.0077	0.01796 ± 0.00176	0.9124 ± 0.0762	0.5181 ± 0.0568	0.20 ± 0.01
5%	SISA†	0.8410 ± 0.0098	0.00000 ± 0.00000	0.9042 ± 0.0621	0.4747 ± 0.0529	2.67 ± 0.09
5%	SISA retraining†	0.8410 ± 0.0098	0.00000 ± 0.00000	0.9042 ± 0.0621	0.4747 ± 0.0529	2.67 ± 0.08
5%	**PALMU**	0.8505 ± 0.0071	0.00085 ± 0.00037	0.8985 ± 0.0608	0.4758 ± 0.0560	0.71 ± 0.02
10%	Full retraining	0.8475 ± 0.0086	0.00000 ± 0.00000	0.9198 ± 0.0745	0.4898 ± 0.0553	2.09 ± 0.03
10%	Labeled-only retraining	0.8339 ± 0.0091	0.07845 ± 0.00373	0.9250 ± 0.0733	0.4868 ± 0.0521	0.15 ± 0.00
10%	Fisher	0.8506 ± 0.0078	0.02391 ± 0.00338	0.8930 ± 0.0650	0.5271 ± 0.0585	0.02 ± 0.00
10%	Fisher + regeneration	0.8506 ± 0.0078	0.02391 ± 0.00338	0.8930 ± 0.0650	0.5271 ± 0.0585	0.16 ± 0.00
10%	SCRUB	0.8507 ± 0.0079	0.02296 ± 0.00328	0.8951 ± 0.0658	0.5257 ± 0.0584	0.14 ± 0.00
10%	SCRUB + regeneration	0.8505 ± 0.0078	0.02240 ± 0.00323	0.8957 ± 0.0665	0.5255 ± 0.0584	0.29 ± 0.00
10%	SalUn	0.8506 ± 0.0078	0.02390 ± 0.00345	0.8929 ± 0.0652	0.5271 ± 0.0586	0.05 ± 0.00
10%	SalUn + regeneration	0.8506 ± 0.0078	0.02389 ± 0.00345	0.8928 ± 0.0653	0.5271 ± 0.0587	0.20 ± 0.00
10%	SISA†	0.8397 ± 0.0079	0.00000 ± 0.00000	0.9185 ± 0.0801	0.4878 ± 0.0573	2.70 ± 0.09
10%	SISA retraining†	0.8397 ± 0.0079	0.00000 ± 0.00000	0.9185 ± 0.0801	0.4878 ± 0.0573	2.71 ± 0.09
10%	**PALMU**	0.8475 ± 0.0086	0.00113 ± 0.00042	0.9197 ± 0.0750	0.4897 ± 0.0555	0.71 ± 0.02
20%	Full retraining	0.8437 ± 0.0114	0.00000 ± 0.00000	0.9445 ± 0.0502	0.5154 ± 0.0349	2.02 ± 0.03
20%	Labeled-only retraining	0.8298 ± 0.0115	0.07893 ± 0.00324	0.9460 ± 0.0483	0.5140 ± 0.0343	0.13 ± 0.01
20%	Fisher	0.8506 ± 0.0078	0.03528 ± 0.00331	0.8940 ± 0.0728	0.5530 ± 0.0364	0.02 ± 0.00
20%	Fisher + regeneration	0.8506 ± 0.0078	0.03528 ± 0.00331	0.8940 ± 0.0728	0.5530 ± 0.0364	0.15 ± 0.01
20%	SCRUB	0.8506 ± 0.0081	0.03389 ± 0.00324	0.8961 ± 0.0726	0.5520 ± 0.0363	0.14 ± 0.00
20%	SCRUB + regeneration	0.8504 ± 0.0084	0.03324 ± 0.00324	0.8966 ± 0.0723	0.5517 ± 0.0362	0.27 ± 0.01
20%	SalUn	0.8504 ± 0.0079	0.03523 ± 0.00329	0.8939 ± 0.0730	0.5530 ± 0.0365	0.05 ± 0.00
20%	SalUn + regeneration	0.8504 ± 0.0079	0.03521 ± 0.00329	0.8940 ± 0.0730	0.5530 ± 0.0365	0.18 ± 0.01
20%	SISA†	0.8361 ± 0.0133	0.00000 ± 0.00000	0.9478 ± 0.0520	0.5161 ± 0.0338	2.64 ± 0.09
20%	SISA retraining†	0.8361 ± 0.0133	0.00000 ± 0.00000	0.9478 ± 0.0520	0.5161 ± 0.0338	2.64 ± 0.10
20%	**PALMU**	0.8438 ± 0.0114	0.00177 ± 0.00040	0.9445 ± 0.0504	0.5155 ± 0.0350	0.69 ± 0.02
30%	Full retraining	0.8405 ± 0.0159	0.00000 ± 0.00000	0.9675 ± 0.0237	0.4990 ± 0.0208	1.94 ± 0.04
30%	Labeled-only retraining	0.8252 ± 0.0149	0.07853 ± 0.00448	0.9654 ± 0.0252	0.4991 ± 0.0222	0.11 ± 0.01
30%	Fisher	0.8506 ± 0.0078	0.04772 ± 0.00685	0.9189 ± 0.0429	0.5405 ± 0.0215	0.02 ± 0.00
30%	Fisher + regeneration	0.8506 ± 0.0078	0.04772 ± 0.00685	0.9189 ± 0.0429	0.5405 ± 0.0215	0.13 ± 0.01
30%	SCRUB	0.8513 ± 0.0081	0.04610 ± 0.00660	0.9209 ± 0.0428	0.5395 ± 0.0214	0.13 ± 0.00
30%	SCRUB + regeneration	0.8508 ± 0.0088	0.04524 ± 0.00640	0.9213 ± 0.0428	0.5393 ± 0.0214	0.25 ± 0.01
30%	SalUn	0.8505 ± 0.0078	0.04771 ± 0.00687	0.9190 ± 0.0431	0.5405 ± 0.0215	0.05 ± 0.00
30%	SalUn + regeneration	0.8505 ± 0.0079	0.04767 ± 0.00685	0.9190 ± 0.0431	0.5405 ± 0.0216	0.16 ± 0.01
30%	SISA†	0.8317 ± 0.0144	0.00000 ± 0.00000	0.9641 ± 0.0185	0.5006 ± 0.0211	2.59 ± 0.16
30%	SISA retraining†	0.8317 ± 0.0144	0.00000 ± 0.00000	0.9641 ± 0.0185	0.5006 ± 0.0211	2.59 ± 0.16
30%	**PALMU**	0.8404 ± 0.0156	0.00336 ± 0.00085	0.9682 ± 0.0234	0.4991 ± 0.0204	0.65 ± 0.01

Table 11: EDM, BERT. Mean ± sample standard deviation over ten seeds. †SISA uses its own sharded reference.

Forget	Method	F1	TV	FS	AUC	Time (s)
5%	Full retraining	0.8108 ± 0.0082	0.00000 ± 0.00000	0.9089 ± 0.0703	0.4826 ± 0.0566	0.57 ± 0.01
5%	Labeled-only retraining	0.8118 ± 0.0069	0.00618 ± 0.00018	0.9081 ± 0.0700	0.4841 ± 0.0574	0.11 ± 0.00
5%	Fisher	0.8174 ± 0.0046	0.02633 ± 0.00390	0.8147 ± 0.1128	0.5879 ± 0.0642	0.01 ± 0.00
5%	Fisher + regeneration	0.8174 ± 0.0046	0.02633 ± 0.00390	0.8147 ± 0.1128	0.5879 ± 0.0642	0.12 ± 0.00
5%	SCRUB	0.8196 ± 0.0041	0.02412 ± 0.00341	0.8200 ± 0.1062	0.5815 ± 0.0666	0.06 ± 0.00
5%	SCRUB + regeneration	0.8196 ± 0.0041	0.02410 ± 0.00340	0.8200 ± 0.1062	0.5815 ± 0.0666	0.17 ± 0.00
5%	SalUn	0.8210 ± 0.0024	0.02527 ± 0.00348	0.8147 ± 0.1079	0.5853 ± 0.0661	0.02 ± 0.00
5%	SalUn + regeneration	0.8210 ± 0.0024	0.02527 ± 0.00348	0.8147 ± 0.1079	0.5853 ± 0.0661	0.13 ± 0.00
5%	SISA†	0.7945 ± 0.0169	0.00000 ± 0.00000	0.9251 ± 0.0512	0.4913 ± 0.0461	1.24 ± 0.14
5%	SISA retraining†	0.7945 ± 0.0169	0.00000 ± 0.00000	0.9251 ± 0.0512	0.4913 ± 0.0461	1.24 ± 0.15
5%	**PALMU**	0.8108 ± 0.0082	0.00019 ± 0.00005	0.9093 ± 0.0701	0.4828 ± 0.0564	0.19 ± 0.00
10%	Full retraining	0.7983 ± 0.0114	0.00000 ± 0.00000	0.9204 ± 0.0599	0.5272 ± 0.0428	0.55 ± 0.02
10%	Labeled-only retraining	0.7973 ± 0.0122	0.00643 ± 0.00018	0.9217 ± 0.0601	0.5271 ± 0.0423	0.10 ± 0.01
10%	Fisher	0.8179 ± 0.0049	0.03831 ± 0.00478	0.7306 ± 0.0788	0.6347 ± 0.0394	0.01 ± 0.00
10%	Fisher + regeneration	0.8179 ± 0.0049	0.03831 ± 0.00478	0.7306 ± 0.0788	0.6347 ± 0.0394	0.11 ± 0.01
10%	SCRUB	0.8160 ± 0.0028	0.03574 ± 0.00441	0.7343 ± 0.0778	0.6329 ± 0.0389	0.06 ± 0.00
10%	SCRUB + regeneration	0.8160 ± 0.0028	0.03570 ± 0.00444	0.7341 ± 0.0780	0.6330 ± 0.0390	0.16 ± 0.01
10%	SalUn	0.8200 ± 0.0026	0.03747 ± 0.00453	0.7286 ± 0.0776	0.6357 ± 0.0388	0.02 ± 0.00
10%	SalUn + regeneration	0.8200 ± 0.0026	0.03747 ± 0.00453	0.7286 ± 0.0776	0.6357 ± 0.0388	0.13 ± 0.01
10%	SISA†	0.7855 ± 0.0153	0.00000 ± 0.00000	0.9166 ± 0.0550	0.5264 ± 0.0437	1.13 ± 0.07
10%	SISA retraining†	0.7855 ± 0.0153	0.00000 ± 0.00000	0.9166 ± 0.0550	0.5264 ± 0.0437	1.13 ± 0.07
10%	**PALMU**	0.7976 ± 0.0111	0.00034 ± 0.00009	0.9203 ± 0.0600	0.5272 ± 0.0429	0.18 ± 0.00
20%	Full retraining	0.7846 ± 0.0185	0.00000 ± 0.00000	0.9679 ± 0.0303	0.5041 ± 0.0223	0.54 ± 0.02
20%	Labeled-only retraining	0.7852 ± 0.0173	0.00656 ± 0.00025	0.9684 ± 0.0309	0.5042 ± 0.0223	0.09 ± 0.00
20%	Fisher	0.8179 ± 0.0049	0.05864 ± 0.00611	0.7606 ± 0.0389	0.6197 ± 0.0194	0.01 ± 0.00
20%	Fisher + regeneration	0.8179 ± 0.0049	0.05864 ± 0.00611	0.7606 ± 0.0389	0.6197 ± 0.0194	0.10 ± 0.00
20%	SCRUB	0.8182 ± 0.0036	0.05618 ± 0.00552	0.7662 ± 0.0396	0.6169 ± 0.0198	0.06 ± 0.00

Forget	Method	F1	TV	FS	AUC	Time (s)
20%	SCRUB + regeneration	0.8175 ± 0.0033	0.05613 ± 0.00550	0.7665 ± 0.0396	0.6167 ± 0.0198	0.15 ± 0.00
20%	SalUn	0.8200 ± 0.0026	0.05814 ± 0.00598	0.7605 ± 0.0394	0.6197 ± 0.0197	0.02 ± 0.00
20%	SalUn + regeneration	0.8200 ± 0.0026	0.05814 ± 0.00598	0.7605 ± 0.0394	0.6197 ± 0.0197	0.12 ± 0.00
20%	SISA†	0.7609 ± 0.0127	0.00000 ± 0.00000	0.9712 ± 0.0271	0.5022 ± 0.0202	1.11 ± 0.08
20%	SISA retraining†	0.7609 ± 0.0127	0.00000 ± 0.00000	0.9712 ± 0.0271	0.5022 ± 0.0202	1.11 ± 0.08
20%	**PALMU**	0.7845 ± 0.0183	0.00072 ± 0.00016	0.9682 ± 0.0306	0.5042 ± 0.0223	0.17 ± 0.00
30%	Full retraining	0.7828 ± 0.0132	0.00000 ± 0.00000	0.9494 ± 0.0367	0.5050 ± 0.0319	0.53 ± 0.03
30%	Labeled-only retraining	0.7816 ± 0.0098	0.00697 ± 0.00043	0.9493 ± 0.0369	0.5048 ± 0.0321	0.11 ± 0.00
30%	Fisher	0.8174 ± 0.0046	0.07958 ± 0.00622	0.7542 ± 0.0394	0.6229 ± 0.0197	0.01 ± 0.00
30%	Fisher + regeneration	0.8174 ± 0.0046	0.07958 ± 0.00621	0.7542 ± 0.0395	0.6229 ± 0.0197	0.11 ± 0.00
30%	SCRUB	0.8178 ± 0.0054	0.07725 ± 0.00558	0.7592 ± 0.0392	0.6204 ± 0.0196	0.06 ± 0.00
30%	SCRUB + regeneration	0.8178 ± 0.0054	0.07712 ± 0.00555	0.7594 ± 0.0392	0.6203 ± 0.0196	0.16 ± 0.00
30%	SalUn	0.8205 ± 0.0026	0.07954 ± 0.00599	0.7551 ± 0.0397	0.6225 ± 0.0198	0.02 ± 0.00
30%	SalUn + regeneration	0.8205 ± 0.0026	0.07953 ± 0.00599	0.7551 ± 0.0397	0.6225 ± 0.0198	0.13 ± 0.00
30%	SISA†	0.7715 ± 0.0093	0.00000 ± 0.00000	0.9491 ± 0.0312	0.5103 ± 0.0291	1.06 ± 0.07
30%	SISA retraining†	0.7715 ± 0.0093	0.00000 ± 0.00000	0.9491 ± 0.0312	0.5103 ± 0.0291	1.06 ± 0.07
30%	**PALMU**	0.7832 ± 0.0128	0.00140 ± 0.00045	0.9493 ± 0.0363	0.5054 ± 0.0318	0.17 ± 0.00

Table 12: AG News, BERT. Mean ± sample standard deviation over ten seeds. †SISA uses its own sharded reference.

Forget	Method	F1	TV	FS	AUC	Time (s)
5%	Full retraining	0.8657 ± 0.0017	0.00000 ± 0.00000	0.8775 ± 0.0834	0.4763 ± 0.0727	0.82 ± 0.04
5%	Labeled-only retraining	0.8628 ± 0.0016	0.05631 ± 0.00072	0.8674 ± 0.0864	0.4761 ± 0.0782	0.06 ± 0.00
5%	Fisher	0.8648 ± 0.0002	0.01553 ± 0.00361	0.8931 ± 0.0867	0.5138 ± 0.0696	0.07 ± 0.04
5%	Fisher + regeneration	0.8648 ± 0.0002	0.01553 ± 0.00361	0.8931 ± 0.0867	0.5138 ± 0.0696	0.16 ± 0.04
5%	SCRUB	0.8650 ± 0.0003	0.01472 ± 0.00349	0.8910 ± 0.0843	0.5118 ± 0.0702	0.18 ± 0.01
5%	SCRUB + regeneration	0.8652 ± 0.0003	0.01424 ± 0.00352	0.8914 ± 0.0844	0.5111 ± 0.0702	0.23 ± 0.01
5%	SalUn	0.8653 ± 0.0003	0.01559 ± 0.00359	0.8930 ± 0.0843	0.5136 ± 0.0689	0.16 ± 0.01
5%	SalUn + regeneration	0.8653 ± 0.0004	0.01562 ± 0.00358	0.8932 ± 0.0839	0.5132 ± 0.0688	0.21 ± 0.01
5%	SISA†	0.8620 ± 0.0022	0.00000 ± 0.00000	0.8732 ± 0.0893	0.4766 ± 0.0765	1.67 ± 0.13
5%	SISA retraining†	0.8620 ± 0.0022	0.00000 ± 0.00000	0.8732 ± 0.0893	0.4766 ± 0.0765	1.71 ± 0.11
5%	**PALMU**	0.8657 ± 0.0016	0.00073 ± 0.00042	0.8772 ± 0.0836	0.4761 ± 0.0728	0.36 ± 0.05
10%	Full retraining	0.8637 ± 0.0017	0.00000 ± 0.00000	0.9450 ± 0.0380	0.5110 ± 0.0326	0.81 ± 0.06
10%	Labeled-only retraining	0.8614 ± 0.0017	0.05694 ± 0.00078	0.9446 ± 0.0337	0.5110 ± 0.0316	0.06 ± 0.00
10%	Fisher	0.8648 ± 0.0002	0.02086 ± 0.00261	0.8972 ± 0.0683	0.5514 ± 0.0342	0.06 ± 0.03
10%	Fisher + regeneration	0.8648 ± 0.0002	0.02086 ± 0.00261	0.8972 ± 0.0683	0.5514 ± 0.0342	0.14 ± 0.04
10%	SCRUB	0.8649 ± 0.0004	0.01980 ± 0.00247	0.9006 ± 0.0677	0.5497 ± 0.0339	0.17 ± 0.01
10%	SCRUB + regeneration	0.8653 ± 0.0004	0.01922 ± 0.00247	0.9018 ± 0.0676	0.5491 ± 0.0338	0.23 ± 0.01
10%	SalUn	0.8652 ± 0.0003	0.02080 ± 0.00252	0.8970 ± 0.0683	0.5515 ± 0.0342	0.16 ± 0.01
10%	SalUn + regeneration	0.8653 ± 0.0002	0.02079 ± 0.00251	0.8972 ± 0.0684	0.5514 ± 0.0342	0.21 ± 0.01
10%	SISA†	0.8608 ± 0.0016	0.00000 ± 0.00000	0.9513 ± 0.0354	0.5099 ± 0.0294	1.61 ± 0.11
10%	SISA retraining†	0.8608 ± 0.0016	0.00000 ± 0.00000	0.9513 ± 0.0354	0.5099 ± 0.0294	1.68 ± 0.13
10%	**PALMU**	0.8639 ± 0.0017	0.00107 ± 0.00036	0.9452 ± 0.0379	0.5110 ± 0.0325	0.36 ± 0.04
20%	Full retraining	0.8611 ± 0.0019	0.00000 ± 0.00000	0.9541 ± 0.0365	0.4804 ± 0.0222	0.80 ± 0.04
20%	Labeled-only retraining	0.8580 ± 0.0030	0.05753 ± 0.00189	0.9503 ± 0.0378	0.4787 ± 0.0232	0.05 ± 0.00
20%	Fisher	0.8648 ± 0.0002	0.03326 ± 0.00316	0.9536 ± 0.0317	0.5194 ± 0.0208	0.05 ± 0.02
20%	Fisher + regeneration	0.8648 ± 0.0002	0.03326 ± 0.00316	0.9536 ± 0.0317	0.5194 ± 0.0208	0.14 ± 0.04
20%	SCRUB	0.8650 ± 0.0003	0.03185 ± 0.00307	0.9559 ± 0.0315	0.5181 ± 0.0207	0.17 ± 0.01
20%	SCRUB + regeneration	0.8651 ± 0.0004	0.03121 ± 0.00314	0.9557 ± 0.0317	0.5180 ± 0.0209	0.22 ± 0.01
20%	SalUn	0.8651 ± 0.0002	0.03317 ± 0.00318	0.9535 ± 0.0318	0.5194 ± 0.0208	0.16 ± 0.01
20%	SalUn + regeneration	0.8651 ± 0.0002	0.03313 ± 0.00320	0.9535 ± 0.0319	0.5194 ± 0.0209	0.21 ± 0.01
20%	SISA†	0.8577 ± 0.0024	0.00000 ± 0.00000	0.9577 ± 0.0351	0.4819 ± 0.0209	1.65 ± 0.13
20%	SISA retraining†	0.8577 ± 0.0024	0.00000 ± 0.00000	0.9577 ± 0.0351	0.4819 ± 0.0209	1.69 ± 0.13
20%	**PALMU**	0.8613 ± 0.0019	0.00205 ± 0.00042	0.9543 ± 0.0366	0.4806 ± 0.0223	0.37 ± 0.06
30%	Full retraining	0.8581 ± 0.0030	0.00000 ± 0.00000	0.9491 ± 0.0223	0.4916 ± 0.0277	0.76 ± 0.04
30%	Labeled-only retraining	0.8545 ± 0.0025	0.05819 ± 0.00156	0.9491 ± 0.0223	0.4899 ± 0.0270	0.04 ± 0.01
30%	Fisher	0.8648 ± 0.0002	0.04121 ± 0.00316	0.9393 ± 0.0493	0.5302 ± 0.0249	0.08 ± 0.03
30%	Fisher + regeneration	0.8648 ± 0.0002	0.04121 ± 0.00316	0.9393 ± 0.0493	0.5302 ± 0.0249	0.13 ± 0.04
30%	SCRUB	0.8650 ± 0.0003	0.03968 ± 0.00314	0.9408 ± 0.0484	0.5291 ± 0.0249	0.17 ± 0.01
30%	SCRUB + regeneration	0.8649 ± 0.0008	0.03875 ± 0.00315	0.9410 ± 0.0481	0.5288 ± 0.0250	0.21 ± 0.01
30%	SalUn	0.8650 ± 0.0002	0.04112 ± 0.00309	0.9394 ± 0.0493	0.5301 ± 0.0249	0.15 ± 0.02
30%	SalUn + regeneration	0.8650 ± 0.0002	0.04104 ± 0.00308	0.9394 ± 0.0493	0.5301 ± 0.0249	0.20 ± 0.01
30%	SISA†	0.8548 ± 0.0021	0.00000 ± 0.00000	0.9497 ± 0.0253	0.4915 ± 0.0280	1.61 ± 0.18
30%	SISA retraining†	0.8548 ± 0.0021	0.00000 ± 0.00000	0.9497 ± 0.0253	0.4915 ± 0.0280	1.59 ± 0.12
30%	**PALMU**	0.8586 ± 0.0031	0.00295 ± 0.00057	0.9491 ± 0.0220	0.4919 ± 0.0277	0.34 ± 0.04

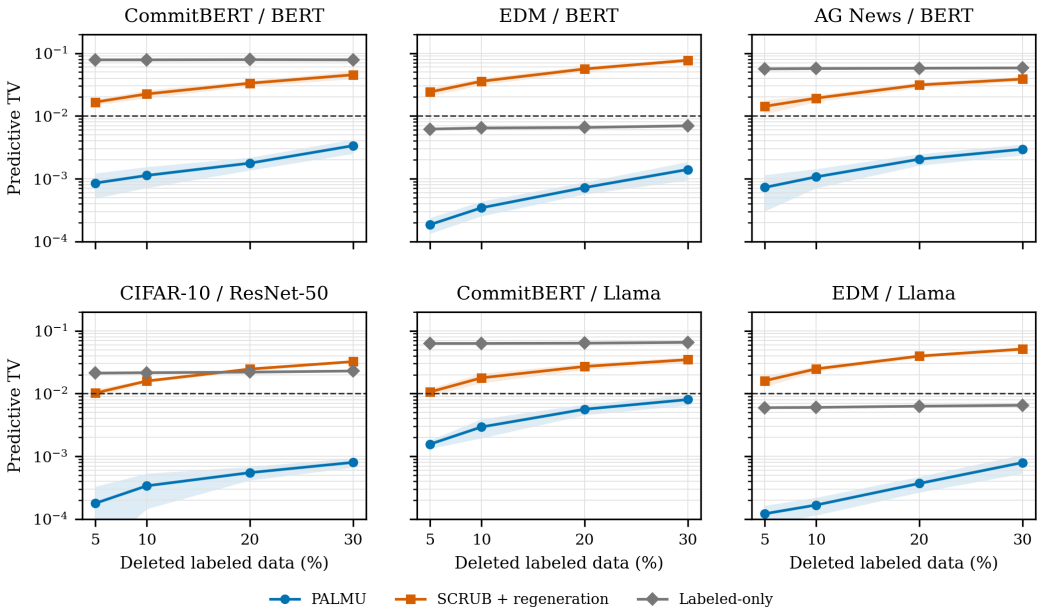

Figure 4: Predictive TV across deletion ratios on four deep fixed representations and two frozen Llama settings. Lines show the mean across seeds 500 to 509, and shaded bands show the sample standard deviation across requests. SCRUB with regeneration represents the tuned baselines, while the tables below report all Fisher, SCRUB, and SalUn variants. The dashed line marks the TV tolerance. The Llama panels cover cached 4,096-dimensional representations, not encoder unlearning.

Table 13: CIFAR-10, ResNet-50. Mean ± sample standard deviation over ten seeds. †SISA uses its own sharded reference.

Forget	Method	F1	TV	FS	AUC	Time (s)
5%	Full retraining	0.8717 ± 0.0011	0.00000 ± 0.00000	0.9190 ± 0.0622	0.4687 ± 0.0412	2.91 ± 0.04
5%	Labeled-only retraining	0.8709 ± 0.0009	0.02119 ± 0.00012	0.9178 ± 0.0623	0.4689 ± 0.0421	0.52 ± 0.02
5%	Fisher	0.8720 ± 0.0002	0.01158 ± 0.00162	0.9414 ± 0.0406	0.5042 ± 0.0367	0.10 ± 0.01
5%	Fisher + regeneration	0.8720 ± 0.0003	0.01158 ± 0.00162	0.9414 ± 0.0406	0.5042 ± 0.0367	0.65 ± 0.03
5%	SCRUB	0.8717 ± 0.0004	0.01037 ± 0.00153	0.9428 ± 0.0429	0.5002 ± 0.0370	0.76 ± 0.03
5%	SCRUB + regeneration	0.8718 ± 0.0004	0.01019 ± 0.00151	0.9430 ± 0.0427	0.5002 ± 0.0369	1.33 ± 0.03
5%	SalUn	0.8720 ± 0.0005	0.01171 ± 0.00145	0.9423 ± 0.0404	0.5039 ± 0.0363	0.28 ± 0.03
5%	SalUn + regeneration	0.8720 ± 0.0005	0.01171 ± 0.00145	0.9423 ± 0.0404	0.5039 ± 0.0363	0.87 ± 0.03
5%	SISA†	0.8689 ± 0.0013	0.00000 ± 0.00000	0.9195 ± 0.0629	0.4717 ± 0.0436	5.21 ± 0.37
5%	SISA retraining†	0.8689 ± 0.0013	0.00000 ± 0.00000	0.9195 ± 0.0629	0.4717 ± 0.0436	5.14 ± 0.35
5%	**PALMU**	0.8717 ± 0.0011	0.00018 ± 0.00015	0.9189 ± 0.0620	0.4686 ± 0.0412	1.02 ± 0.06
10%	Full retraining	0.8711 ± 0.0011	0.00000 ± 0.00000	0.9331 ± 0.0317	0.4665 ± 0.0159	2.85 ± 0.03
10%	Labeled-only retraining	0.8701 ± 0.0011	0.02152 ± 0.00015	0.9317 ± 0.0312	0.4658 ± 0.0156	0.50 ± 0.02
10%	Fisher	0.8720 ± 0.0002	0.01792 ± 0.00098	0.9789 ± 0.0189	0.5006 ± 0.0146	0.11 ± 0.04
10%	Fisher + regeneration	0.8720 ± 0.0002	0.01792 ± 0.00098	0.9789 ± 0.0189	0.5006 ± 0.0146	0.65 ± 0.04
10%	SCRUB	0.8716 ± 0.0002	0.01610 ± 0.00093	0.9779 ± 0.0183	0.4971 ± 0.0145	0.76 ± 0.03
10%	SCRUB + regeneration	0.8716 ± 0.0003	0.01586 ± 0.00091	0.9777 ± 0.0183	0.4970 ± 0.0145	1.30 ± 0.03
10%	SalUn	0.8719 ± 0.0005	0.01774 ± 0.00092	0.9786 ± 0.0190	0.5005 ± 0.0147	0.28 ± 0.02
10%	SalUn + regeneration	0.8719 ± 0.0005	0.01774 ± 0.00092	0.9786 ± 0.0190	0.5005 ± 0.0147	0.83 ± 0.03
10%	SISA†	0.8685 ± 0.0010	0.00000 ± 0.00000	0.9380 ± 0.0307	0.4692 ± 0.0158	5.13 ± 0.41
10%	SISA retraining†	0.8685 ± 0.0010	0.00000 ± 0.00000	0.9380 ± 0.0307	0.4692 ± 0.0158	5.07 ± 0.32
10%	**PALMU**	0.8712 ± 0.0011	0.00034 ± 0.00019	0.9330 ± 0.0317	0.4665 ± 0.0158	0.99 ± 0.02
20%	Full retraining	0.8698 ± 0.0010	0.00000 ± 0.00000	0.9567 ± 0.0263	0.4784 ± 0.0131	2.76 ± 0.05
20%	Labeled-only retraining	0.8686 ± 0.0009	0.02196 ± 0.00033	0.9546 ± 0.0266	0.4773 ± 0.0133	0.48 ± 0.03
20%	Fisher	0.8720 ± 0.0002	0.02698 ± 0.00171	0.9706 ± 0.0159	0.5116 ± 0.0124	0.11 ± 0.04
20%	Fisher + regeneration	0.8720 ± 0.0002	0.02698 ± 0.00171	0.9706 ± 0.0159	0.5116 ± 0.0124	0.61 ± 0.04
20%	SCRUB	0.8717 ± 0.0005	0.02484 ± 0.00167	0.9725 ± 0.0120	0.5091 ± 0.0124	0.75 ± 0.02
20%	SCRUB + regeneration	0.8717 ± 0.0004	0.02457 ± 0.00166	0.9725 ± 0.0120	0.5090 ± 0.0125	1.27 ± 0.03
20%	SalUn	0.8720 ± 0.0003	0.02682 ± 0.00173	0.9706 ± 0.0160	0.5117 ± 0.0123	0.28 ± 0.02
20%	SalUn + regeneration	0.8720 ± 0.0003	0.02681 ± 0.00173	0.9706 ± 0.0160	0.5117 ± 0.0123	0.80 ± 0.03
20%	SISA†	0.8669 ± 0.0013	0.00000 ± 0.00000	0.9618 ± 0.0288	0.4809 ± 0.0144	5.24 ± 0.66
20%	SISA retraining†	0.8669 ± 0.0013	0.00000 ± 0.00000	0.9618 ± 0.0288	0.4809 ± 0.0144	5.23 ± 0.65
20%	**PALMU**	0.8699 ± 0.0010	0.00055 ± 0.00013	0.9567 ± 0.0262	0.4783 ± 0.0131	0.97 ± 0.02

Forget	Method	F1	TV	FS	AUC	Time (s)
30%	Full retraining	0.8674 ± 0.0012	0.00000 ± 0.00000	0.9489 ± 0.0309	0.4745 ± 0.0154	2.70 ± 0.03
30%	Labeled-only retraining	0.8660 ± 0.0016	0.02288 ± 0.00026	0.9467 ± 0.0308	0.4734 ± 0.0154	0.46 ± 0.01
30%	Fisher	0.8720 ± 0.0002	0.03525 ± 0.00151	0.9728 ± 0.0152	0.5086 ± 0.0135	0.10 ± 0.03
30%	Fisher + regeneration	0.8720 ± 0.0002	0.03525 ± 0.00151	0.9728 ± 0.0152	0.5086 ± 0.0135	0.58 ± 0.03
30%	SCRUB	0.8714 ± 0.0006	0.03264 ± 0.00148	0.9734 ± 0.0118	0.5065 ± 0.0136	0.74 ± 0.02
30%	SCRUB + regeneration	0.8716 ± 0.0007	0.03230 ± 0.00146	0.9734 ± 0.0119	0.5065 ± 0.0136	1.23 ± 0.02
30%	SalUn	0.8720 ± 0.0004	0.03510 ± 0.00143	0.9727 ± 0.0151	0.5086 ± 0.0135	0.29 ± 0.03
30%	SalUn + regeneration	0.8720 ± 0.0004	0.03509 ± 0.00143	0.9727 ± 0.0151	0.5086 ± 0.0135	0.79 ± 0.04
30%	SISA†	0.8648 ± 0.0020	0.00000 ± 0.00000	0.9562 ± 0.0308	0.4781 ± 0.0154	5.08 ± 0.27
30%	SISA retraining†	0.8648 ± 0.0020	0.00000 ± 0.00000	0.9562 ± 0.0308	0.4781 ± 0.0154	5.09 ± 0.25
30%	**PALMU**	0.8675 ± 0.0011	0.00080 ± 0.00014	0.9490 ± 0.0309	0.4745 ± 0.0154	0.95 ± 0.02

Table 14: CommitBERT, Llama. Mean $\pm$ sample standard deviation over ten seeds. †SISA uses its own sharded reference.

Forget	Method	F1	TV	FS	AUC	Time (s)
5%	Full retraining	0.9019 ± 0.0065	0.00000 ± 0.00000	0.9381 ± 0.0502	0.4887 ± 0.0394	47.58 ± 0.64
5%	Labeled-only retraining	0.8885 ± 0.0062	0.06286 ± 0.00166	0.9392 ± 0.0434	0.4866 ± 0.0360	3.24 ± 0.05
5%	Fisher	0.9037 ± 0.0056	0.01119 ± 0.00154	0.9224 ± 0.0583	0.5323 ± 0.0369	0.09 ± 0.00
5%	Fisher + regeneration	0.9037 ± 0.0056	0.01119 ± 0.00154	0.9224 ± 0.0583	0.5323 ± 0.0369	3.37 ± 0.05
5%	SCRUB	0.9037 ± 0.0055	0.01101 ± 0.00155	0.9234 ± 0.0580	0.5314 ± 0.0371	0.59 ± 0.00
5%	SCRUB + regeneration	0.9037 ± 0.0057	0.01060 ± 0.00147	0.9248 ± 0.0570	0.5300 ± 0.0372	3.88 ± 0.05
5%	SalUn	0.9038 ± 0.0055	0.01118 ± 0.00154	0.9221 ± 0.0581	0.5322 ± 0.0371	0.22 ± 0.00
5%	SalUn + regeneration	0.9037 ± 0.0055	0.01118 ± 0.00154	0.9222 ± 0.0580	0.5322 ± 0.0371	3.51 ± 0.04
5%	SISA†	0.8947 ± 0.0087	0.00000 ± 0.00000	0.9473 ± 0.0557	0.4950 ± 0.0390	59.35 ± 4.39
5%	SISA retraining†	0.8947 ± 0.0087	0.00000 ± 0.00000	0.9473 ± 0.0557	0.4950 ± 0.0390	59.44 ± 4.35
5%	**PALMU**	0.9020 ± 0.0062	0.00156 ± 0.00024	0.9383 ± 0.0495	0.4896 ± 0.0394	7.63 ± 0.22
10%	Full retraining	0.9007 ± 0.0059	0.00000 ± 0.00000	0.9299 ± 0.0299	0.4980 ± 0.0398	46.64 ± 0.93
10%	Labeled-only retraining	0.8881 ± 0.0054	0.06294 ± 0.00332	0.9413 ± 0.0415	0.4951 ± 0.0369	2.99 ± 0.05
10%	Fisher	0.9037 ± 0.0056	0.01870 ± 0.00338	0.9027 ± 0.0714	0.5483 ± 0.0362	0.09 ± 0.00
10%	Fisher + regeneration	0.9037 ± 0.0056	0.01870 ± 0.00338	0.9027 ± 0.0714	0.5483 ± 0.0362	3.12 ± 0.05
10%	SCRUB	0.9038 ± 0.0057	0.01840 ± 0.00326	0.9043 ± 0.0705	0.5473 ± 0.0361	0.59 ± 0.01
10%	SCRUB + regeneration	0.9035 ± 0.0054	0.01781 ± 0.00309	0.9050 ± 0.0700	0.5466 ± 0.0363	3.63 ± 0.03
10%	SalUn	0.9036 ± 0.0055	0.01871 ± 0.00339	0.9024 ± 0.0713	0.5485 ± 0.0362	0.22 ± 0.00
10%	SalUn + regeneration	0.9036 ± 0.0055	0.01870 ± 0.00339	0.9024 ± 0.0713	0.5485 ± 0.0361	3.26 ± 0.03
10%	SISA†	0.8926 ± 0.0081	0.00000 ± 0.00000	0.9349 ± 0.0336	0.4948 ± 0.0378	57.61 ± 3.40
10%	SISA retraining†	0.8926 ± 0.0081	0.00000 ± 0.00000	0.9349 ± 0.0336	0.4948 ± 0.0378	57.57 ± 3.30
10%	**PALMU**	0.9010 ± 0.0059	0.00294 ± 0.00099	0.9292 ± 0.0290	0.4992 ± 0.0400	7.48 ± 0.28
20%	Full retraining	0.8958 ± 0.0071	0.00000 ± 0.00000	0.9450 ± 0.0388	0.4984 ± 0.0349	44.28 ± 0.42
20%	Labeled-only retraining	0.8816 ± 0.0059	0.06360 ± 0.00245	0.9432 ± 0.0392	0.5005 ± 0.0358	2.51 ± 0.03
20%	Fisher	0.9037 ± 0.0056	0.02820 ± 0.00318	0.8898 ± 0.0729	0.5551 ± 0.0365	0.09 ± 0.01
20%	Fisher + regeneration	0.9037 ± 0.0056	0.02820 ± 0.00318	0.8899 ± 0.0729	0.5551 ± 0.0365	2.66 ± 0.05
20%	SCRUB	0.9038 ± 0.0055	0.02772 ± 0.00307	0.8917 ± 0.0726	0.5542 ± 0.0363	0.58 ± 0.00
20%	SCRUB + regeneration	0.9034 ± 0.0057	0.02703 ± 0.00305	0.8924 ± 0.0733	0.5538 ± 0.0366	3.14 ± 0.03
20%	SalUn	0.9037 ± 0.0055	0.02820 ± 0.00318	0.8898 ± 0.0728	0.5551 ± 0.0364	0.22 ± 0.00
20%	SalUn + regeneration	0.9037 ± 0.0055	0.02820 ± 0.00318	0.8897 ± 0.0728	0.5551 ± 0.0364	2.80 ± 0.03
20%	SISA†	0.8862 ± 0.0084	0.00000 ± 0.00000	0.9468 ± 0.0330	0.4993 ± 0.0325	55.62 ± 2.36
20%	SISA retraining†	0.8862 ± 0.0084	0.00000 ± 0.00000	0.9468 ± 0.0330	0.4993 ± 0.0325	55.64 ± 2.36
20%	**PALMU**	0.8971 ± 0.0066	0.00563 ± 0.00108	0.9449 ± 0.0383	0.5003 ± 0.0348	7.24 ± 0.22
30%	Full retraining	0.8905 ± 0.0076	0.00000 ± 0.00000	0.9559 ± 0.0306	0.5138 ± 0.0237	42.37 ± 0.41
30%	Labeled-only retraining	0.8765 ± 0.0059	0.06551 ± 0.00349	0.9594 ± 0.0307	0.5137 ± 0.0220	2.25 ± 0.04
30%	Fisher	0.9037 ± 0.0056	0.03594 ± 0.00359	0.8534 ± 0.0531	0.5733 ± 0.0266	0.09 ± 0.01
30%	Fisher + regeneration	0.9037 ± 0.0056	0.03594 ± 0.00359	0.8535 ± 0.0531	0.5733 ± 0.0266	2.39 ± 0.05
30%	SCRUB	0.9037 ± 0.0056	0.03545 ± 0.00349	0.8554 ± 0.0533	0.5723 ± 0.0266	0.57 ± 0.00
30%	SCRUB + regeneration	0.9029 ± 0.0056	0.03469 ± 0.00346	0.8560 ± 0.0528	0.5720 ± 0.0264	2.87 ± 0.04
30%	SalUn	0.9037 ± 0.0056	0.03595 ± 0.00357	0.8534 ± 0.0533	0.5733 ± 0.0266	0.22 ± 0.00
30%	SalUn + regeneration	0.9037 ± 0.0056	0.03595 ± 0.00357	0.8534 ± 0.0533	0.5733 ± 0.0266	2.52 ± 0.04
30%	SISA†	0.8788 ± 0.0107	0.00000 ± 0.00000	0.9602 ± 0.0328	0.5135 ± 0.0225	52.99 ± 3.23
30%	SISA retraining†	0.8788 ± 0.0107	0.00000 ± 0.00000	0.9602 ± 0.0328	0.5135 ± 0.0225	53.06 ± 3.22
30%	**PALMU**	0.8932 ± 0.0073	0.00802 ± 0.00153	0.9552 ± 0.0342	0.5160 ± 0.0238	7.02 ± 0.13

Table 15: EDM, Llama. Mean $\pm$ sample standard deviation over ten seeds. †SISA uses its own sharded reference.

Forget	Method	F1	TV	FS	AUC	Time (s)
5%	Full retraining	0.8295 ± 0.0044	0.00000 ± 0.00000	0.9369 ± 0.0394	0.4934 ± 0.0380	4.14 ± 0.04

Forget	Method	F1	TV	FS	AUC	Time (s)
5%	Labeled-only retraining	0.8308 ± 0.0053	0.00596 ± 0.00016	0.9391 ± 0.0368	0.4922 ± 0.0361	0.59 ± 0.00
5%	Fisher	0.8278 ± 0.0000	0.01880 ± 0.00363	0.8518 ± 0.0704	0.5741 ± 0.0352	0.06 ± 0.02
5%	Fisher + regeneration	0.8278 ± 0.0000	0.01880 ± 0.00363	0.8518 ± 0.0704	0.5741 ± 0.0352	0.68 ± 0.03
5%	SCRUB	0.8295 ± 0.0029	0.01609 ± 0.00311	0.8730 ± 0.0706	0.5635 ± 0.0353	0.30 ± 0.03
5%	SCRUB + regeneration	0.8302 ± 0.0031	0.01600 ± 0.00309	0.8730 ± 0.0705	0.5635 ± 0.0353	0.94 ± 0.01
5%	SalUn	0.8291 ± 0.0027	0.01873 ± 0.00375	0.8537 ± 0.0699	0.5732 ± 0.0350	0.14 ± 0.03
5%	SalUn + regeneration	0.8291 ± 0.0027	0.01873 ± 0.00375	0.8537 ± 0.0699	0.5732 ± 0.0350	0.77 ± 0.01
5%	SISA†	0.8186 ± 0.0101	0.00000 ± 0.00000	0.9214 ± 0.0463	0.4917 ± 0.0467	6.81 ± 0.54
5%	SISA retraining†	0.8186 ± 0.0101	0.00000 ± 0.00000	0.9214 ± 0.0463	0.4917 ± 0.0467	6.82 ± 0.53
5%	**PALMU**	0.8295 ± 0.0044	0.00012 ± 0.00004	0.9365 ± 0.0390	0.4932 ± 0.0381	1.32 ± 0.02
10%	Full retraining	0.8280 ± 0.0147	0.00000 ± 0.00000	0.9187 ± 0.0668	0.5018 ± 0.0543	3.98 ± 0.06
10%	Labeled-only retraining	0.8296 ± 0.0138	0.00602 ± 0.00021	0.9186 ± 0.0670	0.5019 ± 0.0544	0.52 ± 0.01
10%	Fisher	0.8278 ± 0.0000	0.02842 ± 0.00261	0.8301 ± 0.1084	0.5849 ± 0.0542	0.06 ± 0.01
10%	Fisher + regeneration	0.8278 ± 0.0000	0.02842 ± 0.00261	0.8301 ± 0.1084	0.5849 ± 0.0542	0.62 ± 0.02
10%	SCRUB	0.8286 ± 0.0021	0.02485 ± 0.00238	0.8467 ± 0.1087	0.5766 ± 0.0543	0.30 ± 0.03
10%	SCRUB + regeneration	0.8286 ± 0.0021	0.02476 ± 0.00237	0.8471 ± 0.1091	0.5765 ± 0.0546	0.87 ± 0.01
10%	SalUn	0.8285 ± 0.0021	0.02826 ± 0.00276	0.8315 ± 0.1094	0.5842 ± 0.0547	0.13 ± 0.03
10%	SalUn + regeneration	0.8285 ± 0.0021	0.02826 ± 0.00276	0.8315 ± 0.1094	0.5842 ± 0.0547	0.71 ± 0.01
10%	SISA†	0.8093 ± 0.0150	0.00000 ± 0.00000	0.9244 ± 0.0539	0.4980 ± 0.0481	6.66 ± 0.35
10%	SISA retraining†	0.8093 ± 0.0150	0.00000 ± 0.00000	0.9244 ± 0.0539	0.4980 ± 0.0481	6.68 ± 0.35
10%	**PALMU**	0.8280 ± 0.0147	0.00017 ± 0.00005	0.9186 ± 0.0667	0.5017 ± 0.0543	1.29 ± 0.02
20%	Full retraining	0.8207 ± 0.0110	0.00000 ± 0.00000	0.9587 ± 0.0317	0.4987 ± 0.0269	3.72 ± 0.04
20%	Labeled-only retraining	0.8217 ± 0.0132	0.00630 ± 0.00023	0.9574 ± 0.0319	0.4986 ± 0.0275	0.38 ± 0.01
20%	Fisher	0.8278 ± 0.0000	0.04415 ± 0.00344	0.8379 ± 0.0480	0.5810 ± 0.0240	0.08 ± 0.02
20%	Fisher + regeneration	0.8278 ± 0.0000	0.04415 ± 0.00344	0.8379 ± 0.0480	0.5810 ± 0.0240	0.49 ± 0.03
20%	SCRUB	0.8293 ± 0.0045	0.03970 ± 0.00276	0.8489 ± 0.0490	0.5755 ± 0.0245	0.28 ± 0.03
20%	SCRUB + regeneration	0.8297 ± 0.0045	0.03959 ± 0.00273	0.8492 ± 0.0491	0.5754 ± 0.0246	0.72 ± 0.01
20%	SalUn	0.8285 ± 0.0021	0.04431 ± 0.00344	0.8373 ± 0.0482	0.5814 ± 0.0241	0.13 ± 0.03
20%	SalUn + regeneration	0.8285 ± 0.0021	0.04431 ± 0.00344	0.8373 ± 0.0482	0.5814 ± 0.0241	0.56 ± 0.01
20%	SISA†	0.8015 ± 0.0219	0.00000 ± 0.00000	0.9589 ± 0.0338	0.5022 ± 0.0274	6.69 ± 0.17
20%	SISA retraining†	0.8015 ± 0.0219	0.00000 ± 0.00000	0.9589 ± 0.0338	0.5022 ± 0.0274	6.70 ± 0.16
20%	**PALMU**	0.8207 ± 0.0110	0.00037 ± 0.00010	0.9586 ± 0.0317	0.4986 ± 0.0269	1.21 ± 0.01
30%	Full retraining	0.8067 ± 0.0227	0.00000 ± 0.00000	0.9700 ± 0.0234	0.5073 ± 0.0181	3.54 ± 0.12
30%	Labeled-only retraining	0.8092 ± 0.0246	0.00651 ± 0.00029	0.9697 ± 0.0232	0.5074 ± 0.0181	0.25 ± 0.00
30%	Fisher	0.8278 ± 0.0000	0.05582 ± 0.00387	0.8225 ± 0.0293	0.5887 ± 0.0146	0.06 ± 0.02
30%	Fisher + regeneration	0.8278 ± 0.0000	0.05582 ± 0.00387	0.8225 ± 0.0293	0.5887 ± 0.0146	0.35 ± 0.03
30%	SCRUB	0.8301 ± 0.0045	0.05134 ± 0.00390	0.8308 ± 0.0283	0.5846 ± 0.0141	0.28 ± 0.02
30%	SCRUB + regeneration	0.8301 ± 0.0045	0.05119 ± 0.00387	0.8309 ± 0.0283	0.5845 ± 0.0142	0.59 ± 0.01
30%	SalUn	0.8285 ± 0.0021	0.05598 ± 0.00413	0.8224 ± 0.0293	0.5888 ± 0.0147	0.14 ± 0.03
30%	SalUn + regeneration	0.8285 ± 0.0021	0.05597 ± 0.00413	0.8224 ± 0.0293	0.5888 ± 0.0147	0.44 ± 0.01
30%	SISA†	0.7890 ± 0.0143	0.00000 ± 0.00000	0.9670 ± 0.0215	0.5110 ± 0.0168	6.25 ± 0.49
30%	SISA retraining†	0.7890 ± 0.0143	0.00000 ± 0.00000	0.9670 ± 0.0215	0.5110 ± 0.0168	6.25 ± 0.50
30%	**PALMU**	0.8071 ± 0.0219	0.00080 ± 0.00026	0.9698 ± 0.0233	0.5075 ± 0.0181	1.16 ± 0.02