Forgetting Has Neighbors: Localized Collateral Forgetting in Machine Unlearning

Polina Dolgova[1,2] **Sebastian U. Stich**[1]
[1]CISPA Helmholtz Center for Information Security [2]Universität des Saarlandes
Saarbrücken, Germany
{polina.dolgova, stich}@cispa.de

Abstract

Machine unlearning aims to remove the influence of selected training examples without full retraining. Standard evaluations often summarize unlearning quality with aggregate metrics, such as accuracy- and forgetting-based scores, which can hide localized failures. We study this failure mode at the example level by comparing the predictions of an unlearned model to those of the model retrained after deletion. We show that this pointwise discrepancy can be highly non-uniform: for gradient-ascent and random-labeling methods, with and without retain-set fine-tuning, it grows with geometric proximity to the forget set. We call this phenomenon *localized collateral forgetting*. Our analysis identifies a mechanism behind the effect: surrogate targets used during unlearning can be inconsistent with the local prediction structure induced by retraining, and this inconsistency propagates through shared representations to nearby examples. Motivated by this mechanism, we propose *Local Teacher Distillation*, a simple mitigation strategy that replaces random targets with soft labels from a small teacher trained only on retained neighbors of the forget set. On CIFAR-100 partial-class deletion, this local teacher brings the unlearned model substantially closer to retraining, especially near the forget set, while maintaining competitive aggregate unlearning metrics.

1 Introduction

Machine unlearning aims to remove the influence of designated training examples from a trained model without retraining the model from scratch. This is needed, for example, when data must be deleted post hoc under regulations such as the GDPR [10]. The gold-standard reference is retraining after deletion: an unlearning method should approximate this reference at lower cost, while preserving utility on non-deleted data and limiting the recoverability of information about the forgotten examples.

Standard evaluations compare unlearned models to retraining using aggregate metrics such as retain accuracy, forget accuracy, test accuracy, and privacy-audit scores. These summaries are useful but incomplete. An unlearned model can appear close to retraining on average while deviating sharply on particular retained or test examples. We therefore study unlearning at the example level, asking where the predictions of an unlearned model differ from those of the retrained reference.

This pointwise view reveals a structured failure mode in common image-classification unlearning methods. Gradient ascent (GA) methods [24] remove the forget set by reversing its training signal, while random-labeling (RL) approaches [25, 11] replace it with corrupted targets. These interventions suppress the forget-set signal, but because examples are coupled through shared representations, the induced perturbation need not remain confined to the deleted examples. It can also affect retained and test points whose representations are close to the forget set.

Figure 1 illustrates this effect on MNIST [17]. For both random labeling and gradient ascent, the gap to retraining increases with proximity to the forgotten class. We call this phenomenon *localized collateral forgetting*: unintended degradation on non-deleted examples, measured relative

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

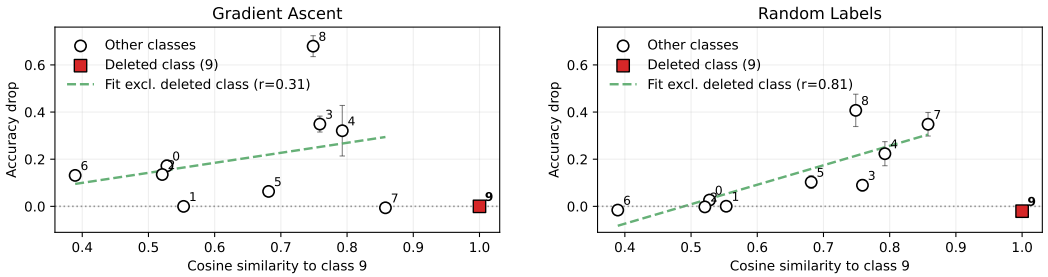

Figure 1: MNIST illustration of *localized collateral forgetting* after deleting class 9, for Gradient Ascent (left) and forget-only Random Labels (right). Non-deleted classes closer to class 9 in representation space show larger accuracy drops relative to retraining. The dashed line is a linear fit over the non-deleted classes only.

to retraining, that concentrates near the forget set in representation space. Retain-set fine-tuning can reduce this degradation, but our analysis shows that forget-set targets inconsistent with retraining can still propagate mismatch to nearby examples.

These observations motivate three questions: what governs the pointwise discrepancy between unlearning and retraining, why is this discrepancy localized, and how can it be reduced? We show theoretically that, for GA and RL-based methods, the discrepancy is governed by alignment with the forget set. Our analysis identifies target mismatch as a key mechanism: corrupted forget-set targets conflict with the local prediction structure induced by retraining. This suggests a direct mitigation strategy: replacing random targets on the forget set with soft labels from a local teacher trained on nearby retained examples.

Our contributions are as follows.

1. **Phenomenon.** We show that collateral forgetting has a localized structure: the discrepancy between an unlearned model and the retrained reference concentrates on non-forget examples that are close to the forget set in representation space.
2. **Theory.** We analyze this localization for gradient ascent, forget-set-only random-label updates, and random-label unlearning with retain-set fine-tuning. The results show that the deviation from retraining is governed by alignment with the forget direction, or more generally by projection onto the forget span. The same analysis identifies target-mismatch as a mechanism underlying localized collateral forgetting.
3. **Method and experiments.** Our analysis suggests a design principle: forget-set targets should be locally consistent with the behavior induced by retraining. We instantiate this principle with *Local Teacher Distillation*, where a small teacher trained on retained neighbors provides soft forget-set targets in place of random labels. On CIFAR-100 partial-class deletion, this method substantially reduces the discrepancy from retraining, especially on affected-class and high-similarity examples, while maintaining competitive aggregate unlearning metrics.

2 Related work

Machine unlearning for image classification. Practical unlearning methods for image classification include gradient ascent (GA) [24], random labeling (RL) [25], fine-tuning (FT) [25], Influence Unlearning (IU) [14, 13], SalUn [11], and AMUN [8]. While these methods are typically evaluated through aggregate metrics, we study a finer failure mode: several unlearning procedures can deviate systematically from retraining on retained and test examples that are close to the forget set.

Retain–forget entanglement. Recent work [28, 3, 9, 18] shows that coupling between retain and forget data can make unlearning harder. We study this coupling at the *point level*, measuring how the unlearning–retraining discrepancy varies across retained and test examples with proximity to the forget set. Moreover, rather than only asking whether the forget set is removed, we identify which nearby non-forget examples are harmed despite being preserved by retraining.

Ripple and collateral effects. Ripple effects [5, 21] and related work on collateral or adjacency effects [26, 23] study original-model side effects on related facts or adjacent concepts. We instead take *retraining* as the reference and study how the resulting discrepancy varies with similarity to the

forget set. This separates benign retraining-induced changes from collateral forgetting: degradation on non-forget examples that retraining would preserve.

Teacher-based unlearning. Prior work uses teacher- or target-based objectives [16, 1, 4]. LTD instead trains its teacher on retained neighbors to approximate the unavailable retraining response; we discuss this distinction in App. A.

3 Collateral Forgetting is Localized

We study a side effect of machine unlearning that we call *collateral forgetting*: the unlearned model unnecessarily degrades predictions on non-forget examples that retraining would preserve. With retraining as the reference, the relevant object is not an absolute prediction change, but the pointwise discrepancy between the unlearned and retrained models. We show that this discrepancy is localized, concentrating on retained and test examples close to the forget set in representation space.

3.1 Theoretical Analysis

We analyze localized collateral forgetting in two stylized settings in which the effect can be explicitly characterized. We begin with gradient-ascent unlearning in linear regression, which provides a simple geometric warm-up, and then turn to random-label unlearning, with and without retain-set fine-tuning, in logistic regression. In each case, our goal is to understand how the unlearning trajectory deviates from retraining after deleting the forget set.

Let the training data be split into a forget set and a retain set,

$$D_f = \{(x_i, y_i)\}_{i \in I_f}, \qquad D_r = \{(x_i, y_i)\}_{i \in I_r},$$

where $I_f \cap I_r = \emptyset$. Let $\ell(\theta; x, y)$ denote the loss, let θ_{full} denote the model trained on the full dataset, and let θ_{retrain} denote the model retrained after deleting D_f. Let $X \in \mathbb{R}^{n \times d}$ denote the design matrix whose rows are the feature vectors $x_i^\top$, and let $y \in \mathbb{R}^n$ be the corresponding vector of responses.

Throughout this section, proximity is measured in representation space rather than input space. In the single-point case, the relevant quantity is the alignment between a query point and the deleted point; with normalized features, this corresponds to cosine similarity up to scale.

Gradient Ascent. We first consider deletion of a single point, $D_f = \{(x_f, y_f)\}$. Starting from θ_{full}, gradient-ascent unlearning on the deleted point is given by

$$\theta_0 = \theta_{\text{full}}, \qquad \theta_{t+1} = \theta_t + \eta \nabla_\theta \ell(\theta_t; x_f, y_f), \qquad t \geq 0,\ \eta > 0.$$

In linear regression, this gives a closed-form characterization of localized collateral forgetting: for a retained example x_q, we measure the effect by the excess squared loss relative to retraining.

Proposition 3.1 (Localized collateral forgetting under gradient ascent)**.** *Let $D_f = \{(x_f, y_f)\}$, and let $X = (x_1, \ldots, x_n)^\top \in \mathbb{R}^{n \times d}$, with $n > d$. Suppose the data follow the linear model*

$$y_i = x_i^\top \theta_{\text{true}} + \xi_i, \qquad \xi_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2),\ x_i \overset{\text{i.i.d.}}{\sim} N(0, \Sigma),\ \Sigma \succ 0,\ \theta_{\text{true}} \in \mathbb{R}^d.$$

Let θ_{full} and θ_{retrain} be the least-squares solutions before and after deleting D_f. Initialize $\theta_0 = \theta_{\text{full}}$, and perform T steps of gradient ascent on the squared loss of the deleted point (x_f, y_f). Then for any retained example $(x_q, y_q) \in D_r$,

$$\mathbb{E}\big[(x_q^\top \theta_T - y_q)^2 \mid X\big] - \mathbb{E}\big[(x_q^\top \theta_{\text{retrain}} - y_q)^2 \mid X\big] = \langle x_q, x_f \rangle^2 \cdot c(T, \eta, x_f)\, \sigma^2 + O_p(1/n),$$

where the expectation is over the responses y, equivalently over the observation noise ξ_i, conditional on X, and $c(T, \eta, x_f)$ does not depend on x_q, and $O_p(1/n)$ term is for fixed T, not uniformly in T.

Proposition 3.1 (for the proof see Appendix B.1) isolates the basic geometry of localized collateral forgetting under gradient ascent. The update on the deleted point affects a retained point only through its overlap with the deleted direction, so the leading-order excess loss scales as $\langle x_q, x_f \rangle^2$. Thus, the discrepancy from retraining is not uniform over the retain set, but concentrates on points that are aligned with the forget point in representation space.

Random Labels. We now turn to random-label unlearning (RL) in logistic regression. In this part, $\ell(\theta; x, y)$ denotes the logistic loss

$$\ell(\theta; x, y) = -\Big(y \log \sigma(\langle \theta, x \rangle) + (1-y) \log(1 - \sigma(\langle \theta, x \rangle))\Big), \qquad \sigma(u) = (1+e^{-u})^{-1}.$$

Here $y \in \{0, 1\}$ for binary labels; for the reference models, θ_{full} and θ_{retrain} are chosen as logit-interpolating parameters, using soft targets $y = \sigma(z) \in [0, 1]$ with the relevant $X\theta = z$.

Write $m = |D_f|$, and let $X_F \in \mathbb{R}^{m \times d}$ denote the rows of X associated with D_f. RL modifies only these examples, replacing their labels by fresh random labels at each step. This makes it a useful case for studying how a surrogate signal on the forget set propagates to nearby retained points.

For each $t \geq 0$, let $\tilde{y}^{(t)} \in \{0, 1\}^m$ be drawn coordinatewise from $\text{Bernoulli}(1/2)$. Starting from θ_{full}, the RL iterates with weight decay $\lambda \geq 0$ are

$$\theta_0 = \theta_{\text{full}}, \qquad \theta_{t+1} = (1 - \eta_t \lambda)\theta_t - \eta_t \sum_{i \in I_f} \nabla_\theta \ell(\theta_t; x_i, \tilde{y}_i^{(t)}), \qquad t \geq 0,\ \eta_t > 0.$$

The next result gives the exact retained-example logit discrepancy between RL and retraining.

Proposition 3.2 (Localized collateral forgetting under Random Labeling)**.** *Assume an interpolating overparameterized setting in which $d \geq n$ and $\text{rank}(X) = n$. For a retained example x_q, write*

$$z_t^{(q)} := \langle \theta_t, x_q \rangle, \qquad z_t^{(F)} := X_F \theta_t, \qquad \Gamma_{0:T} := \prod_{t=0}^{T-1} (1 - \eta_t \lambda).$$

Let

$$\alpha_q := (X_F X_F^\top)^{-1} X_F x_q,$$

so that $X_F^\top \alpha_q$ is the orthogonal projection of x_q onto $\text{span}(X_F^\top)$. Then

$$\langle \theta_T, x_q \rangle - \langle \theta_{\text{retrain}}, x_q \rangle = (\Gamma_{0:T} - 1) z_0^{(q)} + \alpha_q^\top \big(z_T^{(F)} - \Gamma_{0:T} z_0^{(F)}\big).$$

Proposition 3.2 (for the proof see Appendix B.2) shows that the discrepancy has two components: a shrinkage term inherited from weight decay, and a perturbation term created by the random-label updates on the forget set. The latter is transferred to x_q through α_q whose induced vector $X_F^\top \alpha_q$ is the projection of x_q onto the span of the deleted examples. This projection is the multi-point analogue of alignment with a single deleted point: for normalized features, a larger projected component means that x_q is closer to the forget set in representation space.

Thus, whenever the shrinkage term is dominated by the forget-set perturbation, for example when

$$\left|(\Gamma_{0:T} - 1) z_0^{(q)}\right| \leq \varepsilon_T \left|\alpha_q^\top \big(z_T^{(F)} - \Gamma_{0:T} z_0^{(F)}\big)\right|, \qquad \varepsilon_T < 1,$$

the finite-time discrepancy is controlled, up to relative error ε_T, by the component of x_q lying in the forget span. In this sense, the formal projection term captures the intuitive locality effect: retained points closer to the forget set are affected more. This regime is especially relevant in practice, since RL unlearning is typically stopped after a small number of epochs rather than run to convergence.

The limiting behavior is also informative. Under the standard SGD convergence conditions in Assumption B.1, if the random-label dynamics are run to convergence, then

$$\mathbb{E}\left[\langle \theta_T, x_q \rangle\right] \to 0, \qquad \mathbb{E}\left[\langle \theta_T, x_q \rangle - \langle \theta_{\text{retrain}}, x_q \rangle\right] \to -\langle \theta_{\text{retrain}}, x_q \rangle.$$

Hence RL does not converge toward retraining in expectation; it drifts toward the zero-logit predictor induced by random labels. Together, the finite-time and asymptotic views show two aspects of the same failure mode: at finite time, the discrepancy is locally structured by the projection of x_q onto the forget span, while in the limit the mean dynamics are misaligned with the retraining target.

Figure 2 synthetically validates the GA and RL predictions: both the linear-regression MSE discrepancy and the logistic-regression squared-logit discrepancy increase with proximity to D_f.

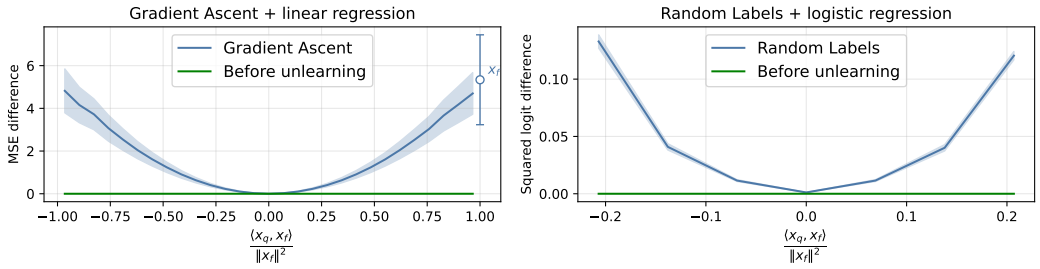

Figure 2: Synthetic validation of the projection-based theory. For GA (left) and forget-only RL (right), retraining-relative discrepancies increase with alignment to the deleted point. RL queries are sampled from $\mathrm{span}(X_R^\top)$.

RL + retain fine-tuning. A natural question is whether fine-tuning on retained data can mitigate this localized degradation. We now add true-label updates on a retained subset $R \subseteq D_r$ to the random-label dynamics on D_f defined above. This captures the basic structure of methods that combine corrupted forget-set updates with retain-set recovery.

Let X_R be the input matrix of R, and let $X_{F\cup R}$ be obtained by stacking X_F and X_R. The random labels are still used on D_f; after taking expectation over this randomness, the induced surrogate objective is

$$L_{\mathrm{RL+FT}}(\theta) := \sum_{(x_i,y_i)\in R} \ell(\theta; x_i, y_i) + \sum_{(x_i,y_i)\in D_f} \ell(\theta; x_i, 1/2) + \frac{\lambda}{2}\|\theta\|^2.$$

Let $\theta^{(\lambda)}_{\mathrm{RL+FT}}$ denote the limiting solution of this surrogate, and let $\{\theta_t\}_{t\geq 0}$ be the corresponding RL+FT stochastic trajectory. Under Assumption B.1, θ_t converges to $\theta^{(\lambda)}_{\mathrm{RL+FT}}$. As in Proposition 3.2, the query dependence is governed by projection onto $\mathrm{span}(X_{F\cup R}^\top)$. Thus, retain fine-tuning changes the relevant subspace, but does not remove the forget directions. We state this explicitly below; we derive the general identity in Appendix B.3 and prove the full-retain statement in Appendix B.4.

Proposition 3.3 (RL with full retain fine-tuning)**.** *Assume the setting of Proposition 3.2 and consider the limit $\lambda \to 0^+$. Suppose $R = D_r$. Then, for every retained training point $(x_q, y_q) \in D_r$,*

$$\lim_{\lambda\to 0^+} \lim_{T\to\infty} \mathbb{E}\left[\langle \theta_T, x_q\rangle - \langle \theta_{\mathrm{retrain}}, x_q\rangle\right] = 0.$$

Moreover, if $\theta_{\mathrm{retrain}}$ is the minimum-norm interpolant on D_r and $x_q \in \mathrm{span}(X^\top)$ with

$$x_q = X_R^\top v_R + X_F^\top v_F,$$

then at the limiting surrogate solution,

$$\lim_{\lambda\to 0^+} \langle \theta^{(\lambda)}_{\mathrm{RL+FT}}, x_q\rangle = \langle \theta_{\mathrm{retrain}}, x_q\rangle - \langle \theta_{\mathrm{retrain}}, X_F^\top v_F\rangle.$$

Thus, for $\lambda \to 0^+$, full retain fine-tuning can restore agreement with retraining on D_r, but this does not imply agreement on general nearby queries. The remaining discrepancy is determined by the component of x_q lying in the forget span. This suggests that mitigating localized collateral forgetting requires improving the signal injected on D_f, rather than relying only on retain-set recovery.

4 Method: Local Teacher Distillation

4.1 Motivation

Target mismatch. The analysis in Section 3 suggests that the main failure mode behind localized collateral forgetting is a mismatch between the targets imposed on D_f and the predictions induced by retraining. Random labels replace the behavior of the retrained model on D_f with arbitrary targets. Since non-forget examples can have nonzero projection onto the feature subspace spanned by the forget examples, this mismatch is transferred to nearby points, causing the localized collateral effect.

Retain fine-tuning only partially addresses this issue. As shown by the RL+FT analysis, fine-tuning on D_r can restore agreement with retraining on retained training points, but it does not generally determine the model's behavior along components in the span of D_f. Thus, query points with representation components in the forget subspace can retain a residual discrepancy. This suggests that the target used on D_f itself must be improved, rather than relying only on retain-set fine-tuning to correct the effect afterward.

Ideal soft labels. The ideal target is the prediction of the retrained model. If $f_{\theta_{\mathrm{retrain}}}$ were available, then for each forget example $x \in D_f$ we would use

$$\hat{y}_{\mathrm{ret}}(x) := f_{\theta_{\mathrm{retrain}}}(x) \in \Delta^{C-1}$$

as a soft label, where Δ^{C-1} is the probability simplex over C classes. Training against $\hat{y}_{\mathrm{ret}}(x)$ would make the forget-set objective locally consistent with retraining, rather than pushing the model toward random targets. However, these labels are unavailable without retraining and are difficult to infer a priori. This is especially clear in partial-class deletion, where the retrained prediction on a deleted example is not determined by its class identity alone, since other same-class examples remain in the training set.

Local approximation. We therefore approximate these ideal soft labels with a small teacher trained on a selected subset of retained examples, rather than on all of D_r. The teacher is used only to produce soft predictions on D_f, which serve as a proxy for $\hat{y}_{\mathrm{ret}}(x)$. The key question is which retained examples should be used to train this teacher.

Support selection. Natural candidates for the teacher support are retained examples near D_f in representation space, as they are most informative about the local behavior of retraining near the forget set. The projection analysis motivates this choice. In the single-point setting $D_f = \{(x_f, y_f)\}$, the induced discrepancy is controlled by the residual left after projecting the forget point x_f onto the support direction x_s:

$$r_s = x_f - P_{x_s} x_f, \qquad \|r_s\|^2 = \|x_f\|^2 \big(1 - \cos^2(x_f, x_s)\big),$$

where $P_{x_s} = x_s x_s^\top / \|x_s\|^2$; see Proposition B.2. Thus, higher cosine similarity yields a smaller residual in this idealized setting. For multiple forget examples, this suggests choosing retained examples close to the forget subspace $\mathrm{span}(X_F^\top)$; in practice, we use nearest-neighbors in representation space.

4.2 Local Support and Teacher-Guided Unlearning

We instantiate the design principle above as *Local Teacher Distillation*. Let $f_{\theta_{\mathrm{full}}}$ be the model trained on the full dataset, with forget set D_f and retain set D_r. The unlearned model is initialized at θ_{full}. Instead of assigning random labels to D_f, we train a small auxiliary teacher on a local subset of retained examples and use its predictions as soft labels on D_f.

Support set selection. Let $h_{\theta_{\mathrm{full}}}(x) \in \mathbb{R}^p$ be the penultimate-layer representation of x, and let $\bar{h}_{\theta_{\mathrm{full}}}(x) = h_{\theta_{\mathrm{full}}}(x)/\|h_{\theta_{\mathrm{full}}}(x)\|_2$. In our partial-class deletion setting, D_f is relatively homogeneous, so we represent it by the average normalized forget embedding

$$\bar{h}_F := \frac{u_F}{\|u_F\|_2}, \quad \text{where} \quad u_F := \sum_{(x_f, y_f) \in D_f} h_{\theta_{\mathrm{full}}}(x_f).$$

We score each retained example by cosine similarity to this average,

$$s(x, D_f) = \left\langle \bar{h}_{\theta_{\mathrm{full}}}(x), \bar{h}_F \right\rangle,$$

and define $S_k(D_f) \subset D_r$ as the k retained examples with largest scores. For heterogeneous forget sets, one can replace the average embedding by per-example or cluster-based nearest-neighbor selection; we discuss this variant in Appendix E.4.

Teacher training. We train a small auxiliary teacher g_ψ on $S_k(D_f)$ using the original hard labels of these retained examples. The teacher is not trained on D_f and does not use forget labels. Its role is to approximate the local prediction structure near the forget set at substantially lower cost than full retraining. In our experiments, the base model is ResNet-56 and g_ψ is ResNet-8. After training, the teacher produces soft labels

$$\hat{y}_\psi(x) := g_\psi(x) \in \Delta^{C-1}, \qquad x \in D_f,$$

Algorithm 1 Local Teacher Distillation for Homogeneous Forget Sets

Input: trained model $f_{\theta_{\text{full}}}$, forget set D_f, retain set D_r, support size k, loss weight β, unlearning epochs E, teacher budget B
Compute normalized embeddings $\bar{h}_{\theta_{\text{full}}}(x)$ for all $x \in D_f \cup D_r$ and $\bar{h}_F$
Select $S_k(D_f) \subset D_r$ as the k retained examples most similar to $\bar{h}_F$
Train a small teacher g_ψ on $S_k(D_f)$ using retained hard labels for budget B
Construct $\widehat{D}_f := \{(x, \hat{y}_\psi(x)) : x \in D_f\}$ from processed teacher predictions
Initialize $\theta \leftarrow \theta_{\text{full}}$
Optimize $\mathcal{L}_{\text{LTD}}(\theta)$ on $D_r \cup \widehat{D}_f$ for E epochs
Return: unlearned model f_θ

where Δ^{C-1} is the probability simplex over C classes. These soft labels are used as proxies for the unavailable retraining soft labels $\hat{y}_{\text{ret}}(x)$. Equivalently, the teacher defines a soft-labeled forget set

$$\widehat{D}_f := \{(x, \hat{y}_\psi(x)) : x \in D_f\}.$$

In practice, we keep and renormalize the top-3 teacher probabilities, which speeds convergence while retaining more than 90% of the prediction mass.

Unlearning objective. Starting from θ_{full}, we optimize

$$\mathcal{L}_{\text{LTD}}(\theta) = \sum_{(x,y)\in D_r} \text{CE}(e_y, f_\theta(x)) + \beta \sum_{(x,\hat{y})\in \widehat{D}_f} \text{CE}(\hat{y}, f_\theta(x)),$$

where $e_y \in \Delta^{C-1}$ is the one-hot label vector, $\beta > 0$ controls the strength of the forget-set distillation term, and $\text{CE}(p, q) := -\sum_{c=1}^{C} p_c \log q_c$. The first term preserves performance on D_r, while the second replaces random labels with teacher soft labels that approximate retraining locally near D_f. In practice, we optimize this objective using minibatches from D_r and $\widehat{D}_f$.

Hyperparameters. Local Teacher Distillation has four main hyperparameters: the support size k, the distillation weight β, the number of unlearning epochs E, and the teacher-training budget B. In the homogeneous-forgetting variant used in our experiments, k is the total number of retained examples selected around the forget set. The weight β controls the strength of the soft-label loss on D_f, E controls the adaptation of the original model after teacher labels are constructed, and B controls how well the auxiliary teacher is fit to the selected support set; in our experiments, we train until a target support-set accuracy is reached, with a maximum epoch budget.

We choose these hyperparameters without access to the retrained model. For the support size k, we use a retrain-free locality diagnostic based on the same similarity structure as support selection. We bin retained examples by similarity, apply one gradient-ascent step on D_f to a copy of the full model, and measure the accuracy drop in each bin relative to the original full model. The largest drops in high-similarity bins identify the neighborhood most affected by forget-set perturbations, which we use to choose k. The remaining parameters are chosen to fit the teacher targets on D_f while preserving performance on D_r. Exact values and ablations are reported in Appendices C.1 and D.

Computational cost. The extra cost of LTD is embedding, support selection, and teacher training. For the homogeneous-forgetting variant used in our partial-class deletion experiments, we represent D_f by the average normalized forget embedding and score retained examples by cosine similarity. After one embedding pass over $D_f \cup D_r$, this costs

$$O(|D_f|p) + O(|D_r|p) + O(|D_r|),$$

for averaging, scoring, and unordered top-k selection using a linear-time selection algorithm, where p is the embedding dimension. The GA-based locality diagnostic adds $O(\text{grad}(D_f)) + O(\text{forward}(D_r)) + O(|D_r|(p+1))$. Teacher training scales with the selected support size k, rather than $|D_r|$, and uses a smaller architecture. The final unlearning stage costs E epochs of standard fine-tuning on $\mathcal{L}_{\text{LTD}}$. Heterogeneous forget set variants are discussed in Appendix E.4.

5 Experiments

Setup. We evaluate our method on CIFAR-100 [15] with a ResNet-56 [12] backbone. We consider partial-deletion regimes, removing 50% and 90% of a single class (we use the '`couch`' class).

Table 1: Main results for class-fraction-to-forget $= 0.5$. Values are mean $\pm$ std over 5 runs; all accuracy metrics are in $\%$. Blue values show deviation from Retrain. Avg. Gap is the mean absolute deviation from Retrain over UA, RA, TA, and MIA.

Method	UA	RA	TA	MIA	Avg. Gap	RTE
Retrain	47.4 ± 3.2	99.9 ± 0.0	72.1 ± 0.2	71.3 ± 3.1	0	1558.9 ± 22.0
RL	73.2 ± 7.1 (+25.8)	94.6 ± 0.8 (-5.3)	67.9 ± 0.6 (-4.2)	84.9 ± 13.6 (+13.6)	12.2	154.4 ± 0.8
FT	79.0 ± 6.8 (+31.6)	94.1 ± 1.1 (-5.8)	67.6 ± 1.0 (-4.5)	28.6 ± 8.3 (-42.7)	21.2	158.6 ± 1.7
GA	68.6 ± 0.6 (+21.2)	98.6 ± 0.0 (-1.3)	69.6 ± 0.0 (-2.5)	65.7 ± 0.3 (-5.6)	7.7	1.8 ± 0.0
IU	70.6 ± 8.1 (+23.2)	99.6 ± 0.0 **(-0.3)**	71.5 ± 0.2 (-0.6)	69.1 ± 7.7 (-2.2)	6.6	29.8 ± 0.5
SalUn	67.2 ± 5.4 (+19.8)	94.3 ± 0.8 (-5.6)	67.9 ± 0.7 (-4.2)	84.0 ± 12.0 (+12.7)	10.6	361.1 ± 6.6
AMUN	67.4 ± 2.2 (+20.0)	94.3 ± 0.1 (-5.6)	72.6 ± 0.1 **(+0.5)**	45.3 ± 3.8 (-26.0)	13.0	171.0 ± 0.8
LTD (ours)	53.4 ± 8.4 **(+6.0)**	99.4 ± 0.1 (-0.5)	70.8 ± 0.3 (-1.3)	69.4 ± 9.4 **(-1.9)**	2.4	226.5 ± 5.2

Additional robustness studies, including 90% deletion, additional affected classes, SVHN [20] with a ViT-Tiny backbone [7], and heterogeneous forget sets, are reported in Appendix E. We refer to the partially deleted class as the *affected class*. Partial-class deletion is a stricter and more realistic setting than whole-class removal: since same-class examples remain in the retained data, retraining behavior near D_f is nontrivial. Hyperparameter ranges and training details for all methods are reported in Appendix C.1. Code is available at `https://github.com/polina-dolgova/local_teacher_distillation`. Results are averaged over 5 runs.

Baselines. We compare against standard unlearning baselines: random-labeling (RL), fine-tuning (FT), gradient ascent (GA), SalUn, Influence Unlearning (IU), and AMUN.

Metrics. We report standard unlearning metrics: retain accuracy (RA), test accuracy (TA), accuracy on D_f (UA), SVC-based membership inference (MIA) [13], Avg. Gap (mean absolute deviation from retrain values), and runtime in seconds (RTE). MIA is included as a standard audit, while our main analysis focuses on *discrepancies from retraining* rather than on optimizing a particular attack metric. To capture localized collateral forgetting, we also report affected-class accuracy on D_r, D_f, and the test subset, isolating localized degradation hidden by aggregate accuracy. Finally, for each bin B induced by the forget-set similarity score $s(x, D_f)$, we measure drops relative to retraining, where zero means agreement with retraining, while larger magnitudes indicate stronger localized deviation:

$$\Delta_{\mathrm{acc}}(B) = \mathrm{Acc}_{\mathrm{retrain}}(B) - \mathrm{Acc}_{\mathrm{unlearn}}(B), \qquad \Delta_{\mathrm{conf}}(B) = \frac{1}{|B|} \sum_{(x,y)\in B} \left[f_{\theta_{\mathrm{retrain}}}(x)_y - f_{\theta_{\mathrm{unlearn}}}(x)_y \right].$$

5.1 Main results

Aggregate and affected-class performance. Following standard empirical unlearning, we compare methods by agreement with retraining, rather than by minimizing accuracy on D_f in isolation. Across aggregate metrics (Table 1), our method achieves the lowest Avg. Gap, indicating the closest approximation to retraining. However, aggregate metrics alone can be misleading. As shown in Table 2 and Figure 4, methods that perform well globally may still induce substantial degradation on affected-class points. In particular, IU—the second-best method by Avg. Gap—exhibits significant drops in $\mathrm{RA}_{\text{affected-class}}$ and $\mathrm{TA}_{\text{affected-class}}$, indicating instability on the retrain-relevant neighborhood. In contrast, our method remains consistently close to retraining both globally and on affected-class subsets, suggesting improved control over localized collateral forgetting. Appendix E shows that the same pattern persists under 90% deletion, across five additional affected classes, and on SVHN with a ViT-Tiny backbone.

Similarity-based analysis. We further analyze these spillover effects by measuring accuracy drop and correct-class confidence drop relative to retraining as a function of similarity to the forget set (Figure 3). We observe a clear locality effect: points with higher similarity to the forget set exhibit substantially larger deviations from retraining, both on the retain and test sets. This confirms that unlearning errors concentrate in the neighborhood of the removed data. Across all similarity bins, our method maintains consistently low deviation from retraining, remaining stable on both datasets. In contrast, competing methods exhibit sharp degradation in high-similarity regions, indicating poor control over localized collateral forgetting.

Ablations. We ablate support selection, teacher architecture, teacher quality, and support size, with full results in Appendix D. Locality is the most important factor: using nearest-neighbor support

Table 2: Results on the subset of points whose true label equals the affected class. Values are mean ± std over 5 runs, in %. Blue values show deviation from Retrain.

Method	$\text{RA}_{\text{affected-class}}$	$\text{UA}_{\text{affected-class}}$	$\text{TA}_{\text{affected-class}}$	Avg. Gap
Retrain	99.7 ± 0.3	47.4 ± 3.1	49.6 ± 5.2	0
RL	88.2 ± 4.0 (-11.4)	73.2 ± 7.1 (+25.8)	50.2 ± 6.0 **(+0.6)**	12.6
FT	95.2 ± 3.6 (-4.5)	79.0 ± 6.8 (+31.6)	58.6 ± 7.9 (+9.0)	15.0
GA	66.6 ± 0.2 (-33.1)	68.6 ± 0.6 (+21.2)	31.0 ± 0.0 (-18.6)	24.3
IU	74.6 ± 5.3 (-25.0)	70.6 ± 8.1 (+23.1)	32.0 ± 3.2 (-17.6)	21.9
SalUn	81.2 ± 5.8 (-18.5)	67.2 ± 5.4 (+19.8)	43.0 ± 6.3 (-6.6)	15.0
AMUN	87.0 ± 1.1 (-12.6)	67.4 ± 2.2 (+19.9)	54.4 ± 2.9 (+4.8)	12.4
LTD (Ours)	97.4 ± 1.4 **(-2.2)**	53.4 ± 8.4 **(+5.9)**	43.0 ± 5.9 (-6.6)	4.9

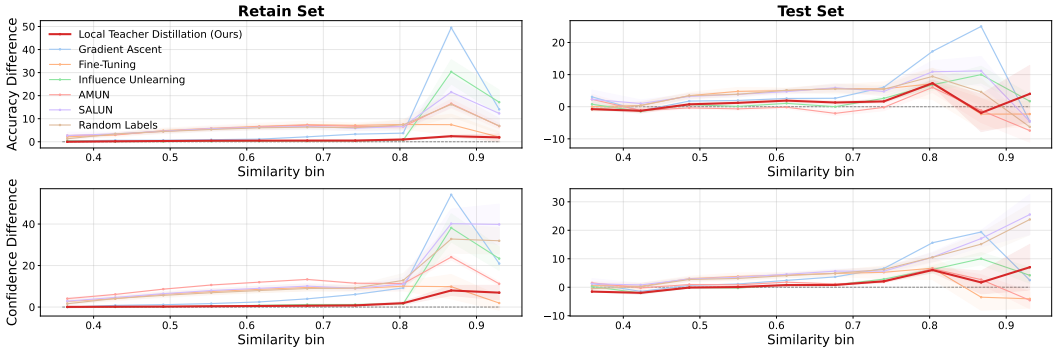

Figure 3: **Localized prediction deviations under 50% class deletion.** We plot bin-level differences from retraining: $\text{Acc}_{\text{retrain}} - \text{Acc}_{\text{unlearn}}$ for accuracy (top row) and $f_{\theta_{\text{retrain}}}(x)_y - f_{\theta_{\text{unlearn}}}(x)_y$ for correct-class confidence (bottom row), as functions of similarity to D_f on both retain and test sets. Larger deviations at high similarity indicate that unlearning effects concentrate near the removed data.

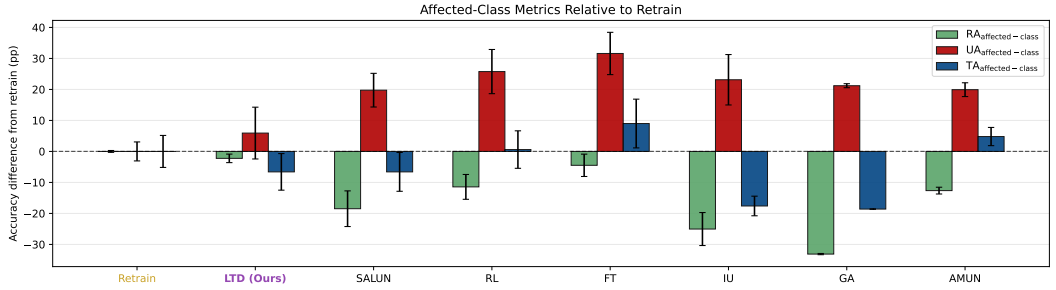

Figure 4: **Affected-class performance difference from Retrain.** Bars show differences from Retrain for $\text{RA}_{\text{affected-class}}$, $\text{UA}_{\text{affected-class}}$, and $\text{TA}_{\text{affected-class}}$; values closer to zero indicate better agreement. The figure reveals localized collateral forgetting not captured by aggregate metrics.

Table 3: **Ablation on support selection.** Nearest-neighbor support yields substantially better alignment with retraining, while random or distant support fails to capture the relevant prediction structure.

	Standard metrics		Forget-class subset metrics			Teacher
Method	**RA**	**TA**	$\text{RA}_{\text{affected-class}}$	$\text{UA}_{\text{affected-class}}$	$\text{TA}_{\text{affected-class}}$	**UA on D_f**
Retrain	99.9 ± 0.0	72.1 ± 0.2	99.7 ± 0.3	47.4 ± 3.1	49.6 ± 5.2	47.4 ± 3.1
Random	99.4 ± 0.0 (-0.5)	70.5 ± 0.2 (-1.5)	97.6 ± 1.3 **(-2.1)**	20.0 ± 3.0 (-27.4)	37.6 ± 3.9 (-12.0)	0.3 ± 0.3 (-47.1)
Farthest	99.4 ± 0.0 (-0.4)	70.6 ± 0.2 (-1.5)	96.2 ± 2.0 (-3.5)	15.9 ± 7.1 (-31.5)	37.8 ± 4.9 (-11.8)	0.0 ± 0.0 (-47.4)
Closest	99.4 ± 0.1 **(-0.4)**	70.8 ± 0.3 **(-1.3)**	97.4 ± 1.4 (-2.2)	53.4 ± 8.4 **(+5.9)**	43.0 ± 5.9 **(-6.6)**	47.92 ± 8.0 **(+0.52)**

substantially improves alignment with retraining compared to random or distant support (Table 3). Additional ablations show that the method is robust to the teacher architecture, while teacher quality and support size affect how well the teacher approximates retraining near the forget set.

6 Conclusion

We studied *localized collateral forgetting*: a discrepancy between unlearning and retraining that concentrates on non-forget examples close to the forget set. Our theoretical analysis shows that this localization arises naturally in gradient-ascent and random-label-based unlearning, and can persist even when retain-set fine-tuning is used as a recovery step.

The analysis identifies target mismatch as a key mechanism: random or corrupted targets on the forget set need not match the local prediction structure induced by retraining. Motivated by this, we proposed *Local Teacher Distillation*, which replaces random targets with soft labels from a small teacher trained on retained neighbors of the forget set. On CIFAR-100 partial-class deletion, LTD reduces discrepancy to retraining while maintaining strong unlearning performance.

Limitations. Our method relies on the quality of the local support set and the representation used to select it. We evaluate controlled image-classification partial-deletion settings designed to isolate localized collateral forgetting; extension to more general architectures, datasets, and deletion regimes remains future work. Our theoretical analysis focuses on stylized linear and logistic settings, providing insight into the underlying mechanism without establishing formal guarantees for deep networks. Finally, although local teachers are trained only on retained examples, support selection relies on representations from the full model and proximity to the forget set, potentially introducing an indirect dependence on forgotten data.

Acknowledgements

This work was supported by the Helmholtz Association's Initiative and Networking Fund on the HAICORE@FZJ partition. We gratefully acknowledge funding from the European Research Council (ERC) under the Horizon Europe Framework Programme (HORIZON) for proposal number 101170430 CollectiveMinds. Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

References

[1] M. Chen, W. Gao, G. Liu, K. Peng, and C. Wang. Boundary unlearning: Rapid forgetting of deep networks via shifting the decision boundary. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7766–7775, 2023. doi: 10.1109/CVPR52729.2023.00750.

[2] Y. Chen. Pytorch cifar models. `https://github.com/chenyaofo/pytorch-cifar-models`. Accessed: 2025-5-17.

[3] J. Cheng, P. Liu, Q. Li, and C. Zhang. Machine unlearning under retain-forget entanglement. *arXiv preprint arXiv:2603.26569*, 2026.

[4] V. S. Chundawat, A. K. Tarun, M. Mandal, and M. Kankanhalli. Can bad teaching induce forgetting? unlearning in deep networks using an incompetent teacher. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 7210–7217, 2023.

[5] R. Cohen, E. Biran, O. Yoran, A. Globerson, and M. Geva. Evaluating the ripple effects of knowledge editing in language models. *Transactions of the Association for Computational Linguistics*, 12:283–298, 2023. URL `https://api.semanticscholar.org/CorpusID:260356612`.

[6] R. Cook and S. Weisberg. *Residuals and Influence in Regression*. Monographs on statistics and applied probability. Chapman and Hall, 1986. URL `https://books.google.de/books?id=aMDpswEACAAJ`.

[7] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020. URL `https://arxiv.org/abs/2010.11929`.

[8] A. Ebrahimpour-Boroojeny, H. Sundaram, and V. Chandrasekaran. Amun: Adversarial machine unlearning, 2025. URL https://arxiv.org/abs/2503.00917.

[9] A. Ebrahimpour-Boroojeny, Y. Wang, and H. Sundaram. On the necessity of output distribution reweighting for effective class unlearning, 2025. URL https://arxiv.org/abs/2506.20893.

[10] European Commission. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance), 2016. URL https://eur-lex.europa.eu/eli/reg/2016/679/oj.

[11] C. Fan, J. Liu, Y. Zhang, E. Wong, D. Wei, and S. Liu. Salun: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=gn0mIhQGNM.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2015. URL https://api.semanticscholar.org/CorpusID:206594692.

[13] J. Jia, J. Liu, P. Ram, Y. Yao, G. Liu, Y. Liu, P. Sharma, and S. Liu. Model sparsity can simplify machine unlearning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=0jZH883i34.

[14] P. W. Koh and P. Liang. Understanding black-box predictions via influence functions. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1885–1894. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/koh17a.html.

[15] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, Toronto, Ontario, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

[16] M. Kurmanji, P. Triantafillou, J. Hayes, and E. Triantafillou. Towards unbounded machine unlearning. *Advances in neural information processing systems*, 36:1957–1987, 2023.

[17] Y. LeCun, C. Cortes, and C. J. Burges. The mnist database of handwritten digits. *http://yann.lecun.com/exdb/mnist*, 1998.

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

[19] D. Montgomery, E. Peck, and G. Vining. *Introduction to Linear Regression Analysis*. Wiley Series in Probability and Statistics. Wiley, 2012. ISBN 9780470542811. URL https://books.google.de/books?id=0yR4KUL4VDkC.

[20] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Ng. Reading digits in natural images with unsupervised feature learning. 2011. URL https://api.semanticscholar.org/CorpusID:16852518.

[21] R. Rinberg, U. Bhalla, I. Shilov, and R. Gandikota. Ripplebench: Capturing ripple effects by leveraging existing knowledge repositories. In *NeurIPS 2025 Workshop on Biosecurity Safeguards for Generative AI*, 2025. URL https://openreview.net/forum?id=rQgPJ3ILLN.

[22] S. P. Singh and D. Alistarh. Woodfisher: Efficient second-order approximation for neural network compression. *Advances in Neural Information Processing Systems*, 33:18098–18109, 2020.

[23] K. Thakral, T. Glaser, T. Hassner, M. Vatsa, and R. Singh. Fine-grained erasure in text-to-image diffusion-based foundation models, 2025. URL https://arxiv.org/abs/2503.19783.

[24] A. Thudi, G. Deza, V. Chandrasekaran, and N. Papernot. Unrolling SGD: Understanding Factors Influencing Machine Unlearning . In *2022 IEEE 7th European Symposium on Security and Privacy (EuroSP)*, pages 303–319, Los Alamitos, CA, USA, June 2022. IEEE Computer Society. doi: 10.1109/EuroSP53844.2022.00027. URL https://doi.ieeecomputersociety.org/10.1109/EuroSP53844.2022.00027.

[25] A. Warnecke, L. Pirch, C. Wressnegger, and K. Rieck. Machine unlearning of features and labels. In *Network and Distributed System Security (NDSS) Symposium 2023: 27 February - 3 March 2023, San Diego, CA, USA*, 2023. ISBN 1-891562-83-5. doi: 10.14722/ndss.2023.23087. 46.23.01; LK 01.

[26] Y. Wen, R. Feng, F. Guo, Y. Wang, R. Le, Y. Song, S. Gao, and S. Shang. Lock on target! precision unlearning via directional control. In C. Christodoulopoulos, T. Chakraborty, C. Rose, and V. Peng, editors, *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 18782–18794, Suzhou, China, Nov. 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.1021. URL https://aclanthology.org/2025.findings-emnlp.1021/.

[27] R. Wightman. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019.

[28] K. Zhao, M. Kurmanji, G.-O. Bărbulescu, E. Triantafillou, and P. Triantafillou. What makes unlearning hard and what to do about it. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=QAbhLBF72K.

Appendix Contents

A Additional Related Work **14**

B Proofs **14**

B.1 Proof of Proposition 3.1 (Localized collateral forgetting under gradient ascent) . . 14

B.2 Proof of Proposition 3.2 (Localized collateral forgetting under random-label unlearning) 16

B.3 General RL+FT identities 18

B.4 Proof of Proposition 3.3 19

B.5 Proof for the Local Support Selection 4.1. 20

C Experimental details **22**

C.1 Hyperparameters 22

D Ablations **23**

D.1 Teacher Type 23

D.2 Teacher versus direct neighbor targets 23

D.3 Teacher Confidence 24

D.4 Number of epochs 26

D.5 Support Size 27

D.5.1 Retrain-free diagnostics 27

D.5.2 Post-hoc comparison to retraining 29

E Additional Experiments **30**

E.1 90% Class Deletion Experiments 30

E.2 Additional affected classes under 50% deletion 32

E.2.1 Class 52: `oak_tree` 32

E.2.2 Class 0: `apple` 33

E.2.3 Class 36: `hamster` 35

E.2.4 Class 68: `road` 36

E.2.5 Class 73: `shark` 37

E.2.6 Summary across classes 39

E.3 Additional Experiments on SVHN with ViT-Tiny 39

E.4 Heterogeneous forget sets 43

E.4.1 Experiments on heterogeneous forget set 43

E.5 Class-Conditioned Analysis 45

F Broader impacts **46**

A Additional Related Work

Teacher-based unlearning. We further discuss the differences between LTD and existing teacher-based unlearning methods, briefly introduced in Section 2. The key distinction is between moving away from the original response and moving toward an estimate of retraining behavior. SCRUB [16] specifies a response to move away from using the full-data teacher, while Bad Teaching [4] uses an incompetent teacher as an explicit forget-set target and Boundary Shrink [1] derives targets from the original decision-boundary geometry. These targets can be poorly matched to partial-class deletion: the Bad Teaching teacher achieves only $\sim 1\%$ accuracy on D_f, compared with 47.4% for retraining and 47.92% for the LTD teacher, while retraining may preserve the original class rather than push every forget example across a decision boundary. LTD instead trains its teacher exclusively on nearby retained examples and uses its soft predictions as a proxy for the unavailable retraining response.

Empirical comparison. We additionally compare LTD with SCRUB [16], Bad Teaching [4], and Boundary Shrink [1] in the same CIFAR-100 partial-class setting. LTD remains substantially closer to retraining, with an overall Avg. Gap of 2.4 versus 13.2, 18.1, and 14.9, respectively, and an affected-class Avg. Gap of 4.9 versus 13.0, 20.4, and 49.2. The baselines fail in different ways: SCRUB shows large deviations on UA and MIA, Bad Teaching strongly over-forgets the deleted subset, and Boundary Shrink causes severe degradation across affected-class splits.

Table 4: Comparison with additional target-based unlearning methods for 50% class deletion. Values are mean $\pm$ std over 5 runs; all accuracy metrics are in %. Blue values show deviation from Retrain. Avg. Gap is the mean absolute deviation from Retrain over UA, RA, TA, and MIA.

Method	UA	RA	TA	MIA	Avg. Gap	RTE
Retrain	47.4 ± 2.9	99.9 ± 0.0	72.1 ± 0.2	71.3 ± 2.8	0	–
SCRUB	71.4 ± 1.9 (+24.0)	99.8 ± 0.0 (-0.1)	71.9 ± 0.1 (-0.2)	42.9 ± 2.5 (-28.4)	13.2	193.9 ± 2.0
Bad Teaching	4.6 ± 2.7 (-42.9)	99.5 ± 0.0 (-0.3)	71.2 ± 0.2 (-0.9)	99.5 ± 0.3 (+28.2)	18.1	219.6 ± 0.9
Boundary Shrink	18.5 ± 0.2 (-29.0)	92.6 ± 0.1 (-7.2)	64.9 ± 0.0 (-7.2)	87.4 ± 0.2 (+16.1)	14.9	1.7 ± 0.3
LTD (ours)	53.4 ± 8.4 (+6.0)	99.4 ± 0.1 (-0.5)	70.8 ± 0.3 (-1.3)	69.4 ± 9.4 (-1.9)	2.4	226.5 ± 5.2

Table 5: Comparison with additional target-based unlearning methods on the affected class for 50% class deletion. Values are mean $\pm$ std over 5 runs, in %. Blue values show deviation from Retrain. Avg. Gap is the mean absolute deviation from Retrain over the three affected-class metrics.

Method	$\text{RA}_{\text{affected-class}}$	$\text{UA}_{\text{affected-class}}$	$\text{TA}_{\text{affected-class}}$	Avg. Gap
Retrain	99.7 ± 0.3	47.4 ± 2.8	49.6 ± 4.7	0
SCRUB	95.3 ± 1.4 (-4.4)	71.4 ± 1.9 (+24.0)	39.0 ± 0.0 (-10.6)	13.0
Bad Teaching	96.3 ± 2.0 (-3.4)	4.6 ± 2.7 (-42.9)	34.6 ± 6.1 (-15.0)	20.4
Boundary Shrink	19.5 ± 0.3 (-80.2)	18.5 ± 0.2 (-29.0)	11.0 ± 0.0 (-38.6)	49.2
LTD (ours)	97.4 ± 1.4 (-2.2)	53.4 ± 8.4 (+5.9)	43.0 ± 5.9 (-6.6)	4.9

B Proofs

B.1 Proof of Proposition 3.1 (Localized collateral forgetting under gradient ascent)

Proof. We initialize $\theta_0 := \theta_{\text{full}}$ and perform gradient ascent updates on the deleted point:

$$\theta_{t+1} = \theta_t + 2\eta\, x_f (x_f^\top \theta_t - y_f).$$

Denote by e^{full} the prediction error of θ_{full}:

$$e_q^{\text{full}} := x_q^\top \theta_{\text{full}} - y_q, \quad e_f^{\text{full}} := x_f^\top \theta_{\text{full}} - y_f.$$

Define the residual at step t:

$$r_t^{(i)} := x_i^\top \theta_t - y_i.$$

Our goal is to compare the squared prediction loss after gradient ascent with the retrained reference:

$$\Delta_{\text{loss}}(x_q) := \mathbb{E}\Big[\big(r_T^{(q)}\big)^2 \mid X\Big] - \mathbb{E}\big[(x_q^\top \theta_{\text{retrain}} - y_q)^2 \mid X\big]\,.$$

We decompose this quantity by adding and subtracting the full-model loss:

$$\Delta_{\text{loss}}(x_q) = \underbrace{\mathbb{E}\Big[\big(r_T^{(q)}\big)^2 \mid X\Big] - \mathbb{E}\Big[\big(r_0^{(q)}\big)^2 \mid X\Big]}_{A_n} + \underbrace{\mathbb{E}\big[(x_q^\top \theta_{\text{full}} - y_q)^2 \mid X\big] - \mathbb{E}\big[(x_q^\top \theta_{\text{retrain}} - y_q)^2 \mid X\big]}_{B_n}.$$

The first term A_n captures the effect of the gradient-ascent updates, while B_n is the correction from the full model to retraining.

Dynamics on the deleted point. We first track how the gradient-ascent update changes the residual on the deleted example. By substituting the update rule, we get

$$r_{t+1}^{(f)} = x_f^\top \theta_{t+1} - y_f = r_t^{(f)} + 2\eta \|x_f\|^2 r_t^{(f)} = (1 + 2\eta\|x_f\|^2) r_t^{(f)}.$$

Thus, gradient ascent amplifies the initial full-model residual on the deleted point:

$$r_t^{(f)} = (1 + 2\eta\|x_f\|^2)^t \, e_f^{\text{full}}.$$

Dynamics on an arbitrary point. We next track how the same update affects a query point x_q. Its residual changes only through the overlap with the deleted direction:

$$r_{t+1}^{(q)} = r_t^{(q)} + 2\eta \langle x_q, x_f \rangle r_t^{(f)}.$$

Substituting the deleted-point dynamics and unrolling the recursion gives

$$r_T^{(q)} = e_q^{\text{full}} + 2\eta \langle x_q, x_f \rangle \sum_{s=0}^{T-1} (1 + 2\eta\|x_f\|^2)^s \, e_f^{\text{full}}.$$

Contribution relative to the full model. Define

$$\alpha_T := 2\eta \langle x_q, x_f \rangle \sum_{r=0}^{T-1} (1 + 2\eta\|x_f\|^2)^r = \frac{\langle x_q, x_f \rangle}{\langle x_f, x_f \rangle} \Big((1 + 2\eta \, \|x_f\|^2)^T - 1 \Big).$$

Then the query residual after T gradient-ascent steps can be written as

$$r_T^{(q)} = e_q^{\text{full}} + \alpha_T e_f^{\text{full}}.$$

Since $\theta_0 = \theta_{\text{full}}$, we have $r_0^{(q)} = e_q^{\text{full}} = x_q^\top \theta_{\text{full}} - y_q$. Hence the change in the conditional squared prediction error relative to the full model is

$$\begin{aligned}
\mathbb{E}\left[\big(r_T^{(q)}\big)^2 \mid X\right] - \mathbb{E}[(x_q^\top \theta_{\text{full}} - y_q)^2 \mid X] &= \mathbb{E}\left[\big(r_T^{(q)}\big)^2 \mid X\right] - \mathbb{E}\left[\big(r_0^{(q)}\big)^2 \mid X\right] \\
&= \mathbb{E}[(e_q^{\text{full}} + \alpha_T e_f^{\text{full}})^2 \mid X] - \mathbb{E}[(e_q^{\text{full}})^2 \mid X] = \alpha_T^2 \mathbb{E}[(e_f^{\text{full}})^2 \mid X] + 2\alpha_T \mathbb{E}[e_f^{\text{full}} e_q^{\text{full}} \mid X].
\end{aligned}$$

Therefore, the first term in the decomposition is

$$A_n = \alpha_T^2 \mathbb{E}\left[(e_f^{\text{full}})^2 \mid X\right] + 2\alpha_T \mathbb{E}\left[e_f^{\text{full}} e_q^{\text{full}} \mid X\right].$$

Asymptotic behavior. We now evaluate A_n and B_n under the linear model. The goal is to show that the leading term comes from the gradient-ascent contribution A_n, while the full-to-retrain correction B_n is lower order.

Let X_{-f} and y_{-f} denote the design matrix and response vector with the deleted example removed. Since $n > d$ and $\Sigma \succ 0$, both $X^\top X$ and $X_{-f}^\top X_{-f}$ are invertible with probability one. In this linear-regression setting,

$$\theta_{\text{full}} = (X^\top X)^{-1} X^\top y, \qquad \theta_{\text{retrain}} = (X_{-f}^\top X_{-f})^{-1} X_{-f}^\top y_{-f}.$$

Conditioning on X, the only randomness is the observation noise. Let $H = X(X^\top X)^{-1} X^\top$ and $h_{ij} = H_{ij}$. It is well-known from standard OLS analysis [19, Sec. 4.2.2]

$$\mathbb{E}[e_i^{\text{full}} e_j^{\text{full}} \mid X] = \sigma^2 (\mathbf{1}\{i = j\} - h_{ij}).$$

Thus,

$$\mathbb{E}[(e_f^{\text{full}})^2 \mid X] = \sigma^2(1 - h_{ff}), \quad \mathbb{E}[e_f^{\text{full}} e_q^{\text{full}} \mid X] = -\sigma^2 h_{fq}.$$

By the law of large numbers, as the sample size n grows,

$$\frac{1}{n} X^\top X \xrightarrow{p} \Sigma,$$

which implies $h_{ij} = O_p(1/n)$ for fixed i, j. Therefore,

$$A_n = \alpha_T^2 \mathbb{E}[(e_f^{\text{full}})^2 \mid X] + 2\alpha_T \mathbb{E}[e_f^{\text{full}} e_q^{\text{full}} | X] = \alpha_T^2 \sigma^2 + O_p(1/n).$$

For B_n, by the standard case-deletion identity for OLS [6], we have

$$x_q^\top \theta_{\text{retrain}} - y_q = e_q^{\text{full}} + \frac{h_{qf}}{1 - h_{ff}} e_f^{\text{full}}.$$

Therefore,

$$B_n = \mathbb{E}\big[(x_q^\top \theta_{\text{full}} - y_q)^2 \mid X\big] - \mathbb{E}\big[(x_q^\top \theta_{\text{retrain}} - y_q)^2 \mid X\big] = \frac{\sigma^2 h_{qf}^2}{1 - h_{ff}} = O_p(1/n^2),$$

Combining the asymptotic expansions of A_n and B_n, we obtain

$$\Delta_{\text{loss}}(x_q) = \alpha_T^2 \sigma^2 + O_p(1/n) + O_p(1/n^2) = \langle x_q, x_f \rangle^2 \left(\frac{(1 + 2\eta \, \|x_f\|^2)^T - 1}{\|x_f\|^2} \right)^2 \sigma^2 + O_p(1/n),$$

where

$$c(T, \eta, x_f) := \left(\frac{(1 + 2\eta \, \|x_f\|^2)^T - 1}{\|x_f\|^2} \right)^2$$

does not depend on x_q, and the B_n term is lower order and is absorbed into the $O_p(1/n)$ remainder. □

B.2 Proof of Proposition 3.2 (Localized collateral forgetting under random-label unlearning)

Assumption B.1 (Conditions for convergence of RL iterates)**.** Let $D_f = \{(x_i, y_i)\}_{i \in I_f}$ be fixed and finite, and let $\tilde{y} = (\tilde{y}_i)_{i \in I_f}$, where $\tilde{y}_i \sim \text{Bernoulli}(1/2)$ independently. Define

$$F(\theta) = \mathbb{E}_{\tilde{y}} \left[\sum_{i \in I_f} \ell(\theta; x_i, \tilde{y}_i) \right] + \frac{\lambda}{2} \|\theta\|^2.$$

Assume:

1. $\lambda > 0$, so that F is strongly convex;
2. the step sizes satisfy
$$\sum_{t=0}^{\infty} \eta_t = \infty, \qquad \sum_{t=0}^{\infty} \eta_t^2 < \infty;$$
3. the stochastic gradients
$$g_t(\theta) := \sum_{i \in I_f} \nabla_\theta \ell(\theta; x_i, \tilde{y}_i^{(t)}) + \lambda \theta$$
are unbiased,
$$\mathbb{E}[g_t(\theta) \mid \theta] = \nabla F(\theta),$$
and have bounded second moment,
$$\mathbb{E}\big[\|g_t(\theta)\|^2 \mid \theta\big] \le C\big(1 + \|\theta\|^2\big)$$
for some constant $C > 0$.

Proof of the proposition.

Proof. The gradient descent step on D_f with randomized labels can be written as:

$$\theta_{t+1} = (1 - \eta_t \lambda)\theta_t - \eta_t \, X_F^\top \big(\sigma(z_t^{(F)}) - \tilde{y}^{(t)}\big).$$

Let

$$G := X_F X_F^\top, \qquad k_q := X_F x_q.$$

Taking inner products gives

$$z_{t+1}^{(F)} = (1 - \eta_t \lambda) z_t^{(F)} - \eta_t \, G\big(\sigma(z_t^{(F)}) - \tilde{y}^{(t)}\big),$$
$$z_{t+1}^{(q)} = (1 - \eta_t \lambda) z_t^{(q)} - \eta_t \, k_q^\top \big(\sigma(z_t^{(F)}) - \tilde{y}^{(t)}\big).$$

Matrix G is invertible by $\operatorname{rank}(X) = n$, and we obtain

$$z_{t+1}^{(q)} - (1 - \eta_t \lambda) z_t^{(q)} = \alpha_q^\top \big(z_{t+1}^{(F)} - (1 - \eta_t \lambda) z_t^{(F)}\big),$$

where

$$\alpha_q := G^{-1} k_q$$

are the coordinates of the orthogonal projection of x_q onto $\operatorname{span}(X_F^\top)$ in the basis given by the rows of X_F.

Unrolling the recursion from $t = 0$ to $T - 1$ and using the fact that $\theta_0 = \theta_{\text{full}}$ yields

$$\langle \theta_T, x_q \rangle - \langle \theta_{\text{full}}, x_q \rangle = (\Gamma_{0:T} - 1) z_0^{(q)} + \alpha_q^\top \big(z_T^{(F)} - \Gamma_{0:T} z_0^{(F)}\big).$$

For the full and retrained references, the retained target logits are the same. Since both references interpolate these logits, $X_R \theta_{\text{full}} = X_R \theta_{\text{retrain}} = z_R$. Hence, for $x_q \in D_r$, $\langle \theta_{\text{full}}, x_q \rangle = \langle \theta_{\text{retrain}}, x_q \rangle$, and therefore

$$\langle \theta_T, x_q \rangle - \langle \theta_{\text{retrain}}, x_q \rangle = (\Gamma_{0:T} - 1) z_0^{(q)} + \alpha_q^\top \big(z_T^{(F)} - \Gamma_{0:T} z_0^{(F)}\big).$$

□

Variance. Under the setting of Proposition 3.2, the only randomness in $\langle \theta_T, x_q \rangle - \langle \theta_{\text{retrain}}, x_q \rangle$ comes from the random labels used in the unlearning dynamics. Therefore,

$$\operatorname{Var}(\langle \theta_T, x_q \rangle - \langle \theta_{\text{retrain}}, x_q \rangle) = \alpha_q^\top \operatorname{Cov}(z_T^{(F)}) \, \alpha_q,$$

This shows that the stochastic variability of RL is not uniform across retained points. It is filtered through the same projection coefficients α_q as the mean discrepancy: points with a larger component in the forget span can exhibit larger variance in their logit discrepancy.

Finite-time regime. If

$$\left|(\Gamma_{0:T} - 1) z_0^{(q)}\right| \le \varepsilon_T \left|\alpha_q^\top \big(z_T^{(F)} - \Gamma_{0:T} z_0^{(F)}\big)\right|, \qquad \varepsilon_T < 1,$$

then the discrepancy is controlled up to relative error ε_T by the forget-set perturbation term. In particular, its dependence on x_q is through the projection coefficients α_q.

Asymptotic behavior. Under Assumption B.1, the RL iterates are stochastic-gradient iterates for the expected random-label objective

$$F(\theta) = \mathbb{E}_{\tilde{y}} \left[\sum_{i \in I_f} \ell(\theta; x_i, \tilde{y}_i) \right] + \frac{\lambda}{2} \|\theta\|^2.$$

Since $\mathbb{E}[\tilde{y}_i] = 1/2$, this objective is equivalently

$$F(\theta) = \sum_{i \in I_f} \ell(\theta; x_i, 1/2) + \frac{\lambda}{2} \|\theta\|^2.$$

Moreover,

$$\nabla F(\theta) = X_F^\top \left(\sigma(X_F \theta) - \frac{1}{2} \mathbf{1} \right) + \lambda \theta.$$

Thus $\nabla F(0) = 0$, and since $\lambda > 0$, F is strongly convex, so its unique minimizer is $\theta^\star = 0$. By standard SGD convergence, $\theta_t \to 0$ under Assumption B.1. Hence, for any fixed query x_q,

$$\langle \theta_t, x_q \rangle \to 0,$$

and, under the corresponding L^1 convergence implied by the same bounded-moment conditions,

$$\mathbb{E}\langle \theta_t, x_q \rangle \to 0.$$

Therefore,

$$\mathbb{E}\left[\langle \theta_T, x_q \rangle - \langle \theta_{\text{retrain}}, x_q \rangle\right] \to -\langle \theta_{\text{retrain}}, x_q \rangle.$$

Relation to the single-point case. In the single-point setting, the logit dynamics is governed by the scalar coefficient

$$\alpha_q = \frac{\langle x_q, x_f \rangle}{\langle x_f, x_f \rangle}.$$

In the multi-point case, this coefficient is replaced by the vector

$$\alpha_q = (X_F X_F^\top)^{-1} X_F x_q,$$

which can be interpreted as the coordinates of the projection of x_q onto $\text{span}(X_F^\top)$.

Thus, the scalar similarity $\langle x_q, x_f \rangle$ is replaced by a projection onto the subspace spanned by the deleted points.

B.3 General RL+FT identities

Let $A = F \cup R$, and let $X_A := X_{F \cup R}$. Define

$$z_t^{(A)} := X_A \theta_t, \qquad G_A := X_A X_A^\top, \qquad k_{q,A} := X_A x_q, \qquad \alpha_{q,A} := G_A^{-1} k_{q,A}.$$

Then $X_A^\top \alpha_{q,A}$ is the projection of x_q onto $\text{span}(X_A^\top)$.

The RL+FT update can be written as

$$\theta_{t+1} = (1 - \eta_t \lambda)\theta_t - \eta_t X_A^\top g_t,$$

where g_t contains the logistic-gradient factors on R with true labels and on D_f with random labels. Hence

$$z_{t+1}^{(q)} - (1 - \eta_t \lambda) z_t^{(q)} = \alpha_{q,A}^\top \left(z_{t+1}^{(A)} - (1 - \eta_t \lambda) z_t^{(A)} \right).$$

Unrolling gives

$$z_T^{(q)} = \Gamma_{0:T} z_0^{(q)} + \alpha_{q,A}^\top \left(z_T^{(A)} - \Gamma_{0:T} z_0^{(A)} \right).$$

Therefore, for any query point x_q,

$$\langle \theta_T, x_q \rangle - \langle \theta_{\text{retrain}}, x_q \rangle = (\Gamma_{0:T} - 1) z_0^{(q)} + \alpha_{q,A}^\top \left(z_T^{(A)} - \Gamma_{0:T} z_0^{(A)} \right) + \langle \theta_{\text{full}} - \theta_{\text{retrain}}, x_q \rangle.$$

For retained training points, the last term vanishes under the interpolation assumption.

Mean dynamics. Taking expectation over the random labels on D_f replaces them by the soft target $1/2$. Thus the mean dynamics converge, under the corresponding SGD convergence conditions, to a minimizer of

$$L_{\text{RL+FT}}(\theta) = \sum_{(x_i, y_i) \in R} \ell(\theta; x_i, y_i) + \sum_{(x_i, y_i) \in D_f} \ell(\theta; x_i, 1/2) + \frac{\lambda}{2} \|\theta\|^2.$$

We denote this minimizer by $\theta_{\text{RL+FT}}$. In general,

$$\mathbb{E}\left[\langle \theta_T, x_q \rangle\right] \to \langle \theta_{\text{RL+FT}}^{(\lambda)}, x_q \rangle,$$

which need not equal $\langle \theta_{\text{retrain}}, x_q \rangle$.

B.4 Proof of Proposition 3.3

Proof. Assume $R = D_r$. We first show that

$$\lim_{\lambda \to 0^+} \lim_{T \to \infty} \mathbb{E}\left[\langle \theta_T^{(\lambda)}, x_q \rangle - \langle \theta_{\text{retrain}}, x_q \rangle\right] = 0.$$

Under Assumption B.1, for every fixed $\lambda > 0$, the expected RL+FT objective admits a unique minimizer θ_λ, and the standard mean-square convergence result for strongly convex SGD with Robbins-Monro step sizes implies

$$\mathbb{E}|\theta_T^{(\lambda)} - \theta_\lambda|^2 \to 0 \qquad \text{as } T \to \infty.$$

Therefore, for every fixed retained query x_q,

$$|\mathbb{E}[\langle \theta_T^{(\lambda)} - \theta_\lambda, x_q \rangle]| \le |x_q| \sqrt{\mathbb{E}|\theta_T^{(\lambda)} - \theta_\lambda|^2} \to 0 \qquad (T \to \infty).$$

Now we analyze θ_λ as $\lambda \to 0^+$. Let

$$z_R = X_R \theta_{\text{retrain}}, \qquad b = (0_F, z_R),$$

where z_R contains the retraining logits on D_r, and b is the target-logit vector consisting of zero logits on D_f and retraining logits on D_r. By the interpolation and full-row-rank assumptions, there exists $\bar{\theta}$ such that $X\bar{\theta} = b$. If Φ denotes the unregularized expected surrogate loss in logit space, optimality of θ_λ gives

$$0 \le \Phi(X\theta_\lambda) - \Phi(b) \le \frac{\lambda}{2}|\bar{\theta}|^2.$$

Therefore, $\lim_{\lambda \to 0^+} \Phi(X\theta_\lambda) = \Phi(b)$. Since b is the unique minimizer of Φ, this implies,

$$X_R \theta_\lambda \to z_R, \qquad X_F \theta_\lambda \to 0 \qquad (\lambda \to 0^+).$$

In particular, for every retained training point x_q,

$$\langle \theta_\lambda, x_q \rangle \to \langle \theta_{\text{retrain}}, x_q \rangle \qquad (\lambda \to 0^+).$$

Combining the two steps above, for every fixed $\lambda > 0$ we have

$$\lim_{T \to \infty} \mathbb{E}[\langle \theta_T^{(\lambda)}, x_q \rangle - \langle \theta_{\text{retrain}}, x_q \rangle] = \langle \theta_\lambda, x_q \rangle - \langle \theta_{\text{retrain}}, x_q \rangle.$$

Here, θ_{retrain} is fixed and does not depend on either T or λ. Taking the outer limit $\lambda \to 0^+$ and using

$$\langle \theta_\lambda, x_q \rangle \longrightarrow \langle \theta_{\text{retrain}}, x_q \rangle$$

gives

$$\lim_{\lambda \to 0^+} \lim_{T \to \infty} \mathbb{E}[\langle \theta_T^{(\lambda)}, x_q \rangle - \langle \theta_{\text{retrain}}, x_q \rangle] = \lim_{\lambda \to 0^+} (\langle \theta_\lambda, x_q \rangle - \langle \theta_{\text{retrain}}, x_q \rangle) = 0.$$

Now assume that θ_{retrain} is the minimum-norm interpolating solution on D_r and that $x_q \in \text{span}(X^\top)$. Write

$$x_q = X_R^\top v_R + X_F^\top v_F.$$

Since $R = D_r$, in the limit $\lambda \to 0^+$ the surrogate solution agrees with retraining on X_R, while the random-label part imposes zero logits on X_F in expectation. Therefore,

$$\lim_{\lambda \to 0^+} \langle \theta_{\text{RL+FT}}^{(\lambda)}, x_q \rangle = \langle \theta_{\text{retrain}}, X_R^\top v_R \rangle.$$

Using

$$\langle \theta_{\text{retrain}}, x_q \rangle = \langle \theta_{\text{retrain}}, X_R^\top v_R \rangle + \langle \theta_{\text{retrain}}, X_F^\top v_F \rangle,$$

we obtain

$$\lim_{\lambda \to 0^+} \langle \theta_{\text{RL+FT}}^{(\lambda)}, x_q \rangle = \langle \theta_{\text{retrain}}, x_q \rangle - \langle \theta_{\text{retrain}}, X_F^\top v_F \rangle.$$

□

Thus, full retain fine-tuning can restore agreement with retraining on retained training points, but this does not imply agreement on general query or test points. For such points, the component aligned with the forget span can still create a discrepancy, as shown by the term $\langle \theta_{\text{retrain}}, X_F^\top v_F \rangle$. Hence retain-set recovery alone is insufficient: mitigating localized collateral forgetting requires improving the signal injected on D_f itself.

B.5 Proof for the Local Support Selection 4.1.

Proposition B.2. *Under the same setting as Proposition 3.3, let x_f be the deleted point and $x_s \in R$ a support point used to construct a pseudo-label for x_f. Let θ_{retrain} be the retrained solution. For fixed $\lambda > 0$, let $\theta^{(\lambda)}_{\text{unlearn}}$ be the limiting solution of the corresponding pseudo-label unlearning objective, and set*

$$\theta_{\text{unlearn}} := \lim_{\lambda \to 0^+} \theta^{(\lambda)}_{\text{unlearn}}.$$

Then for any query point x_q,

$$\Delta := |\langle \theta_{\text{unlearn}}, x_q \rangle - \langle \theta_{\text{retrain}}, x_q \rangle| = \left| \alpha_{q,F} \left(\theta_{\text{retrain}}\right)^\top (P_{x_s} x_f - x_f) \right|,$$

where $P_{x_s} = x_s x_s^\top / \|x_s\|^2$.

Corollary B.3. *Let*

$$r_s := x_f - P_{x_s} x_f.$$

Then

$$\Delta = \left| \alpha_{q,F} \left(\theta_{\text{retrain}}\right)^\top r_s \right| \le |\alpha_{q,F}| \, \|\theta_{\text{retrain}}\| \, \|r_s\|.$$

Moreover,

$$\|r_s\|^2 = \|x_f\|^2 \big(1 - \cos^2(x_f, x_s)\big).$$

Thus, support points with larger cosine similarity to x_f yield a smaller upper bound on Δ. In particular, nearest neighbors of x_f in cosine similarity provide a principled choice of support examples for constructing the teacher model.

Proof. Let

$$X_R := [x_1 \quad \cdots \quad x_{f-1} \quad x_{f+1} \quad \cdots \quad x_{n-1}]^\top, \qquad X_{F \cup R} := \begin{bmatrix} x_f^\top \\ X_R \end{bmatrix} \in \mathbb{R}^{n \times d}.$$

Then

$$k_{q,F\cup R} = X_{F\cup R} x_q, \quad G_{F\cup R} = X_{F\cup R} X_{F\cup R}^\top.$$

The coefficients of the orthogonal projection of x_q onto $\operatorname{span}(X_{F\cup R}^\top)$ are given by

$$\alpha_q = G_{F\cup R}^{-1} k_{q,F\cup R}.$$

Helper model trained on x_s. Since the helper model is trained only on x_s and we consider the minimum-norm solution, it is collinear with x_s, hence there exists $\beta \in \mathbb{R}$ such that

$$\theta_s^* = \beta \, x_s.$$

Let θ_{full} be the minimum-norm logistic regression solution trained on the full dataset D. Matching logits on x_s gives

$$\langle \theta_s^*, x_s \rangle = \beta \, x_s^\top x_s = \langle \theta_{\text{full}}, x_s \rangle = x_s^\top \theta_{\text{full}},$$

so

$$\beta = \frac{x_s^\top \theta_{\text{full}}}{\|x_s\|^2}, \qquad \theta_s^* = \frac{x_s^\top \theta_{\text{full}}}{\|x_s\|^2} x_s.$$

Transferred logits. Applying the helper model to F:

$$z_\infty^{(F)} = \langle \theta_s^*, x_f \rangle = \frac{x_s^\top \theta_{\text{full}}}{\|x_s\|^2} \, x_s^\top x_f.$$

For R (true labels are preserved):

$$z_\infty^{(R)} = X_R \theta_{\text{full}}$$

Thus,

$$z_\infty^{(F\cup R)} = \begin{bmatrix} z_\infty^{(F)} \\ z_\infty^{(R)} \end{bmatrix}.$$

Prediction on x_q.

$$\alpha_q^\top z_\infty^{(F\cup R)} = \alpha_{q,F}^\top z_\infty^{(F)} + \alpha_{q,R}^\top z_\infty^{(R)},$$

where

$$\alpha_q = \begin{bmatrix}\alpha_{q,F}\\ \alpha_{q,R}\end{bmatrix}, \qquad \alpha_{q,F}\in\mathbb{R}^{|F|}, \quad \alpha_{q,R}\in\mathbb{R}^{|R|},$$

with $\alpha_{q,F}$ containing the first $|F|$ coordinates of α_q, and $\alpha_{q,R}$ containing the remaining $|R|$ coordinates.

Substituting:

$$\alpha_q^\top z_\infty^{(F\cup R)} = \alpha_{q,F}^\top \frac{x_s^\top \theta_{\text{full}}}{\|x_s\|^2}\, x_s^\top x_f + \alpha_{q,R}^\top X_R \theta_{\text{full}}$$

Deviation from the retrained model. By definition, we have

$$\Delta := |\langle \theta_{\text{unlearn}}, x_q\rangle - \langle \theta_{\text{retrain}}, x_q\rangle| = |\alpha_q^\top z_\infty^{(F\cup R)} - \langle \theta_{\text{retrain}}, x_q\rangle|.$$

Since the coordinates of α_q are the coefficients of the orthogonal projection of x_q onto $\operatorname{span}\{D\}$, and $\theta_{\text{retrain}} \in \operatorname{span}\{R\}$, we have

$$\begin{aligned}\langle \theta_{\text{retrain}}, x_q\rangle &= \langle \theta_{\text{retrain}}, X_{F\cup R}^\top \alpha_q\rangle = \langle \theta_{\text{retrain}}, x_f\alpha_{q,F} + X_R^\top \alpha_{q,R}\rangle \\ &= \alpha_{q,F}^\top x_f^\top \theta_{\text{retrain}} + \alpha_{q,R}^\top X_R \theta_{\text{retrain}} = \alpha_{q,F}^\top \left(P_{X_R^\top} x_f\right)^\top \theta_{\text{full}} + \alpha_{q,R}^\top X_R \theta_{\text{full}},\end{aligned}$$

where $P_{X_R^\top} x_f := \Pi_{\operatorname{span}(X_R^\top)}(x_f)$.

Then

$$\begin{aligned}\Delta &= |\alpha_q^\top z_\infty^{(F\cup R)} - \langle \theta_{\text{retrain}}, x_q\rangle| \\ &= \left|\left(\alpha_{q,F}^\top \frac{x_s^\top \theta_{\text{full}}}{\|x_s\|^2}\, x_s^\top x_f + \alpha_{q,R}^\top X_R \theta_{\text{full}}\right) - \left(\alpha_{q,F}^\top \left(P_{X_R^\top} x_f\right)^\top \theta_{\text{full}} + \alpha_{q,R}^\top X_R \theta_{\text{full}}\right)\right| \\ &= \left|\alpha_{q,F}^\top \left(\frac{x_s^\top \theta_{\text{full}}}{\|x_s\|^2}\, x_s^\top x_f - \left(P_{X_R^\top} x_f\right)^\top \theta_{\text{full}}\right)\right| \\ &= \left|\alpha_{q,F}^\top \theta_{\text{full}}^\top \left(\frac{x_s x_s^\top}{\|x_s\|^2}\, x_f - P_{X_R^\top} x_f\right)\right| = \left|\alpha_{q,F}^\top \theta_{\text{full}}^\top \left(P_{x_s} x_f - P_{X_R^\top} x_f\right)\right| \\ &= \left|\alpha_{q,F} \theta_{\text{retrain}}^\top \left(P_{x_s} x_f - x_f\right)\right|,\end{aligned}$$

where we used $P_{x_s} x_f \in \operatorname{span}(X_R^\top)$ because $x_s \in R$, and $\theta_{\text{retrain}} = P_R \theta_{\text{full}}$ for the minimum-norm retrained interpolant.

Let

$$r_s := x_f - P_{x_s} x_f.$$

Then

$$\Delta = |\alpha_{q,F}\, \theta_{\text{retrain}}^\top r_s|.$$

Hence, the deviation depends on three factors: the coefficient $\alpha_{q,F}$, the retrained parameter vector θ_{retrain}, and the residual r_s of approximating x_f by the one-dimensional direction x_s.

Moreover,

$$\|r_s\|^2 = \|x_f - P_{x_s} x_f\|^2 = \|x_f\|^2 - \|P_{x_s} x_f\|^2 = \|x_f\|^2 \big(1 - \cos^2(x_f, x_s)\big),$$

so the norm of the residual is small when x_s is well aligned with x_f. At the same time, the deviation is not determined by $\|r_s\|$ alone, but by its interaction with θ_{retrain} through the inner product $\theta_{\text{retrain}}^\top r_s$. Thus, even for a small residual, the deviation may vary depending on how strongly this residual correlates with the retrained solution.

Importantly, $\alpha_{q,F}$ depends only on the geometry of x_f, X_R, and x_q, and does not depend on the choice of x_s. In particular, selecting x_s close to x_f affects only the residual term and cannot make $\alpha_{q,F}$ larger. Note that for $x_q \in R$ one has $\alpha_{q,F} = 0$, whereas for unseen test points $x_q \notin \operatorname{span}(R)$ this coefficient is generally nonzero. □

C Experimental details

C.1 Hyperparameters

Hyperparameter selection. Hyperparameters are selected without access to retraining, using the same rule for all methods. High retain accuracy is a useful retrain-free sanity check in this setting, since overparameterized image-classification models typically fit the retained training data well both before and after retraining. Among configurations with affected-class retain accuracy above 70% and overall retain accuracy above 90%, we choose the one with the lowest accuracy on D_f; if none satisfy these constraints, we choose the one with the highest overall retain accuracy. For LTD, using the teacher accuracy on D_f as a proxy target leads to the same selected regime, close to the best RA/teacher-UA trade-off.

Main experiments (CIFAR-100). For the full and retrained models, we use the ResNet-56 architecture from the PyTorch CIFAR Models repository and follow its CIFAR training protocol [2]. Specifically, we train for 200 epochs with SGD, batch size 256, learning rate 0.1, momentum 0.9, weight decay $5 \cdot 10^{-4}$, Nesterov momentum, and cosine learning-rate decay to zero. We select the checkpoint with the best test accuracy.

For GA, FT, RL, SalUn, and AMUN, we perform the unlearning updates with SGD using momentum 0.9 and weight decay 10^{-6}, following the corresponding public implementations. For Influence Unlearning, we use the implementation released with the SalUn paper.

We tune the main method-specific hyperparameters as follows. For GA, we use 5 unlearning epochs and tune the learning rate in $[10^{-6}, 10^{-2}]$. For FT and RL, we use 20 unlearning epochs and tune the learning rate in $[10^{-3}, 10^{-1}]$. For Influence Unlearning, we tune the WoodFisher [22] inverse-Hessian scaling parameter α in $[1, 20]$. For SalUn, we use 20 unlearning epochs, tune the learning rate in $[10^{-3}, 10^{-1}]$, and tune the mask ratio in $[0.1, 0.9]$. For AMUN, we use 20 unlearning epochs, tune the learning rate in $[10^{-3}, 10^{-1}]$, and evaluate both the standard and `advonly` variants.

For our method, we use 20 unlearning epochs and tune the learning rate in $[10^{-6}, 10^{-3}]$. We set $\beta = 2$, selected from $[1, 5]$ using early training dynamics, to ensure that the forget-set distillation signal is not overwhelmed by retain-set updates. Following the retrain-free support-size diagnostic in Appendix D.5.1, we use a support set of size 1000, which approximately corresponds to selecting examples above a similarity threshold of 0.75; the same threshold regime works for the additional affected classes. We train a ResNet-8 local teacher with SGD until it reaches 0.99 training accuracy. The support size and teacher-accuracy threshold are justified by the ablations. For the final student update, we use AdamW rather than SGD, since the goal is to move quickly toward the teacher-provided soft targets on the forget set. To control for optimizer effects, we also tested AdamW for the baseline methods (GA, FT, RL, SalUn, and AMUN), but this did not improve their results; in most cases, performance was worse than with their standard SGD-based updates.

MNIST. For the MNIST illustration, we delete class 9 and average results over 10 random repeats. We use 300 unlearning steps for RL and 120 steps for GA, with learning rate $5 \cdot 10^{-5}$ and weight decay 10^{-6} for both methods. Error bars show 95% confidence intervals for the class-wise accuracy drop, computed across random repeats using the Student-t interval.

Synthetic illustrations. For the linear-regression GA illustration, we delete one point and evaluate the MSE shift on the retain set. We use $n_{\text{train}} = 5000$, $d = 10$, noise variance $\sigma^2 = 0.1$, 450 unlearning steps, and unlearning learning rate 0.05. Results are averaged over 100 independent runs.

For the logistic-regression RL illustration, we delete one point and sample $n_{\text{test}} = 10000$ query inputs from $\text{span}(X_R^\top)$, where X_R is the retained-input matrix. The corresponding query logits are defined using the same linear combinations of the retained logits. We use $n_{\text{train}} = 15$, $n_{\text{delete}} = 1$, $d = 20$, 500 unlearning steps, and unlearning learning rate 0.01. Results are averaged over 100 independent runs.

Additional experiments (SVHN, ViT-Tiny). For the full and retrained models we use ViT-Tiny [7] (`vit_tiny_patch16_224` from `timm` [27]), adapted for 32×32 inputs with patch size 4 (yielding 64 patch tokens). We train from scratch for 100 epochs with AdamW (lr$= 10^{-3}$, weight decay 0.05) and cosine learning-rate decay to zero, batch size 256, selecting the checkpoint with the best validation accuracy. For the baseline unlearning methods we search the learning rate within the same ranges as in the CIFAR-100 experiments, using 10 unlearning epochs instead. For our method, we

use 10 unlearning epochs with AdamW (lr$= 10^{-4}$) and forget-loss weight $\beta = 3$. The ResNet-8 local teacher is trained with SGD (lr$= 0.005$, cosine decay) until it reaches 0.99 training accuracy. For 50% deletion we use a support set of size $n_s = 4500$; for 90% we use $n_s = 1200$.

Compute resources. All experiments were run on a single NVIDIA A100 GPU with `num_workers=4` for data loading. Per-run wall-clock runtimes are reported as RTE in the experimental tables.

Existing assets. We use CIFAR-100 [15], MNIST [17], and the ResNet-56 implementation from PyTorch CIFAR Models [2], which is released under the BSD-3-Clause license. We access these models through the public repository and do not redistribute modified versions. We also use public baseline implementations where available and cite the corresponding works.

D Ablations

We ablate four components of the proposed method: support selection (reported in the main text), teacher architecture, teacher quality, and support size. Each ablation is reported as a separate subsection to isolate the contribution of the corresponding design choice.

D.1 Teacher Type

We study the effect of the teacher model class. We compare lightweight image-space models (ResNet-8 and a shallow CNN), the full model (ResNet-56), and an MLP trained on frozen embeddings of the full model. This ablation evaluates how the choice of model family and representation affects approximation to retraining.

Table 6: Teacher model ablation: standard metrics and runtime. Values are mean ± std over 5 runs.

Method	RA	TA	RTE
Retrain	99.9 ± 0.0	72.1 ± 0.2	1548.8 ± 11.6
Small CNN	99.4 ± 0.2 **(-0.4)**	70.7 ± 0.2 (-1.3)	224.6 ± 6.1
ResNet-8	99.4 ± 0.1 **(-0.4)**	70.8 ± 0.3 (-1.3)	226.5 ± 5.2
ResNet-56	99.5 ± 0.1 **(-0.4)**	70.8 ± 0.3 (-1.3)	361.6 ± 31.6
MLP Frozen	99.5 ± 0.1 **(-0.4)**	71.0 ± 0.5 **(-1.1)**	191.2 ± 2.1

Table 7: Teacher model ablation: affected-class metrics and teacher accuracy on D_f. Values are mean ± std over 5 runs.

Method	$\text{RA}_{\text{affected-class}}$	$\text{UA}_{\text{affected-class}}$	$\text{TA}_{\text{affected-class}}$	Teacher UA on D_f
Retrain	99.7 ± 0.3	47.4 ± 3.1	49.6 ± 5.2	47.4 ± 3.1
Small CNN	98.8 ± 0.9 (-0.9)	54.4 ± 6.4 (+7.0)	46.6 ± 2.9 **(-3.0)**	43.2 ± 9.6 (-4.2)
ResNet-8	97.4 ± 1.4 (-2.2)	53.4 ± 8.4 **(+5.9)**	43.0 ± 5.9 (-6.6)	47.9 ± 8.0 **(+0.5)**
ResNet-56	97.5 ± 1.2 (-2.2)	53.3 ± 6.5 **(+5.9)**	41.8 ± 6.5 (-7.8)	42.6 ± 9.0 (-4.8)
MLP Frozen	99.6 ± 0.4 **(-0.1)**	99.4 ± 0.2 (+52.0)	65.2 ± 4.2 (+15.6)	98.5 ± 0.4 (+51.1)

We observe in the Tables 6–7 that the choice of teacher architecture within image-space models has only a minor effect on performance: small CNNs, ResNet-8, and ResNet-56 achieve comparable results across both aggregate and affected-class metrics. At the same time, smaller models are significantly more efficient to train, offering a favorable trade-off between accuracy and runtime.

In contrast, the MLP trained on frozen embeddings fails to forget, exhibiting near-perfect accuracy on the forget set and large deviations from retraining. This is expected, as the embedding representation is produced by the full model and already encodes the forgotten data, preventing effective unlearning.

D.2 Teacher versus direct neighbor targets

We additionally test whether the local teacher is necessary, or whether it is sufficient to construct forget-set targets directly from nearby retained examples. For this ablation, we keep the same locality

principle but remove the teacher training step. Instead, the neighbor-label variants construct soft targets for D_f directly from the labels of nearby retained examples, using neighborhoods of size $k = 500$ or $k = 1000$. Thus, this comparison isolates the role of the teacher: both approaches use local retained information, but LTD first trains a small model on the selected support set and then uses its predictions as soft targets on D_f.

Tables 8–9 show that direct neighbor labels preserve overall RA and TA reasonably well, but do not match the retraining behavior on D_f. In particular, the neighbor variants substantially overestimate UA compared to retraining, which leads to larger aggregate and affected-class Avg. Gap. Increasing the neighborhood size from $k = 500$ to $k = 1000$ improves alignment, but the gap remains considerably larger than for LTD.

These results suggest that locality alone is not sufficient. Directly using nearby retained labels gives a useful local signal, but it does not capture the retraining-induced prediction structure as well as a trained local teacher. The teacher appears to act as a smoother local approximation: it uses the same neighboring retained data, but converts them into soft targets that better match the behavior of the retrained model near the forget set.

Table 8: Main results for the neighbor-label variants. Values are mean ± std; All accuracy metrics are in %. Blue values show deviation from Retrain. Avg. Gap is the mean absolute deviation from Retrain over UA, RA, TA, and MIA.

Method	UA	RA	TA	MIA	Avg. Gap	RTE
Retrain	47.4 ± 3.2	99.9 ± 0.0	72.1 ± 0.2	71.3 ± 3.1	0	1723.9 ± 38.6
Neighbor ($k = 500$)	83.3 ± 4.2 (+35.9)	99.6 ± 0.1 (-0.3)	71.1 ± 0.2 (-1.0)	94.7 ± 2.2 (+23.4)	15.1	186.8 ± 0.7
Neighbor ($k = 1000$)	67.1 ± 7.4 (+19.6)	99.5 ± 0.1 (-0.3)	70.7 ± 0.3 (-1.3)	99.5 ± 0.4 (+28.2)	12.4	185.8 ± 1.6
LTD (ours)	53.4 ± 8.4 (+6.0)	99.4 ± 0.1 (-0.5)	70.8 ± 0.3 (-1.3)	69.4 ± 9.4 (-1.9)	2.4	226.5 ± 5.2

Table 9: Results on the affected-class subset for the neighbor-label variants. Values are mean ± std; Retrain is averaged over 5 runs. All accuracy metrics are in %. Blue values show deviation from Retrain. Avg. Gap is computed over $\text{RA}_{\text{affected-class}}$, $\text{UA}_{\text{affected-class}}$, and $\text{TA}_{\text{affected-class}}$.

Method	$\text{RA}_{\text{affected-class}}$	$\text{UA}_{\text{affected-class}}$	$\text{TA}_{\text{affected-class}}$	Avg. Gap
Retrain	99.7 ± 0.3	47.4 ± 3.1	49.6 ± 5.2	0
Neighbor ($k = 500$)	98.8 ± 0.0 (-0.9)	83.3 ± 4.2 (+35.9)	46.7 ± 1.7 (-2.9)	13.2
Neighbor ($k = 1000$)	98.1 ± 1.1 (-1.5)	67.1 ± 7.4 (+19.6)	42.0 ± 4.2 (-7.6)	9.6
LTD (Ours)	97.4 ± 1.4 (-2.2)	53.4 ± 8.4 (+5.9)	43.0 ± 5.9 (-6.6)	4.9

D.3 Teacher Confidence

In our algorithm, the teacher is trained on the support set until it reaches a target training accuracy threshold on the support set. We vary this threshold and study its effect on unlearning quality, localized collateral forgetting mitigation, and runtime (Figures 5– 8).

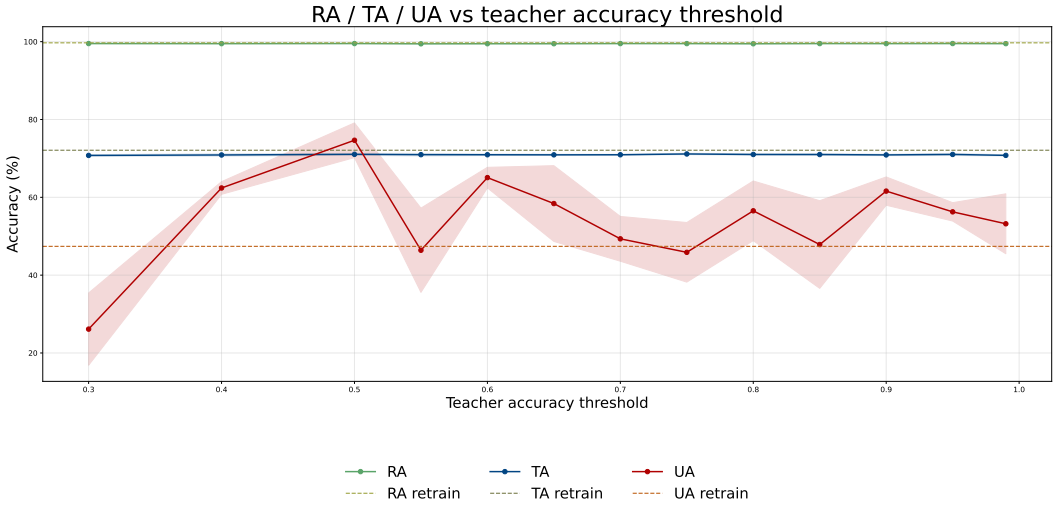

Figure 5: Standard metrics (RA, TA, UA) as a function of the teacher accuracy threshold. Performance remains stable across the considered range.

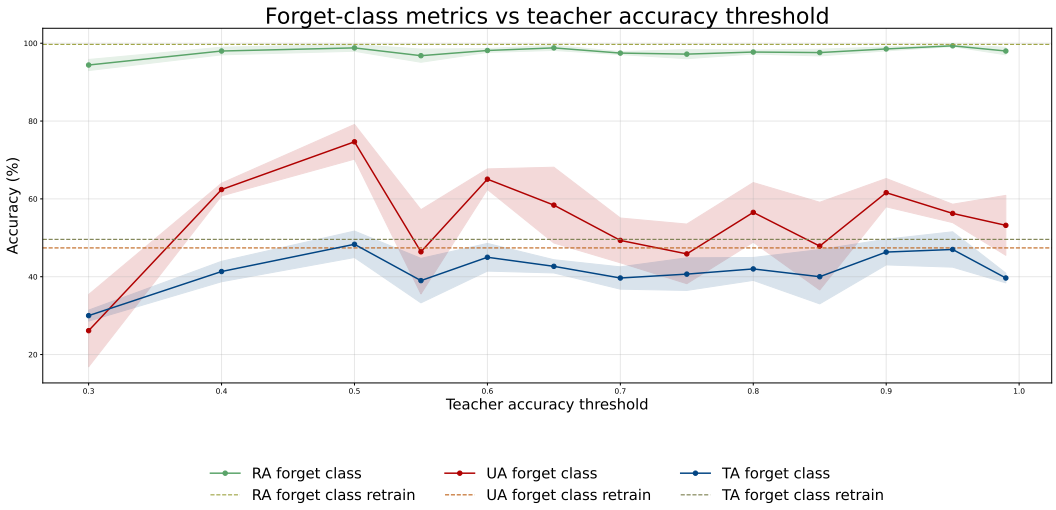

Figure 6: Affected-class metrics as a function of the teacher accuracy threshold. Metrics quickly saturate, indicating that moderate teacher accuracy is sufficient to approximate retraining.

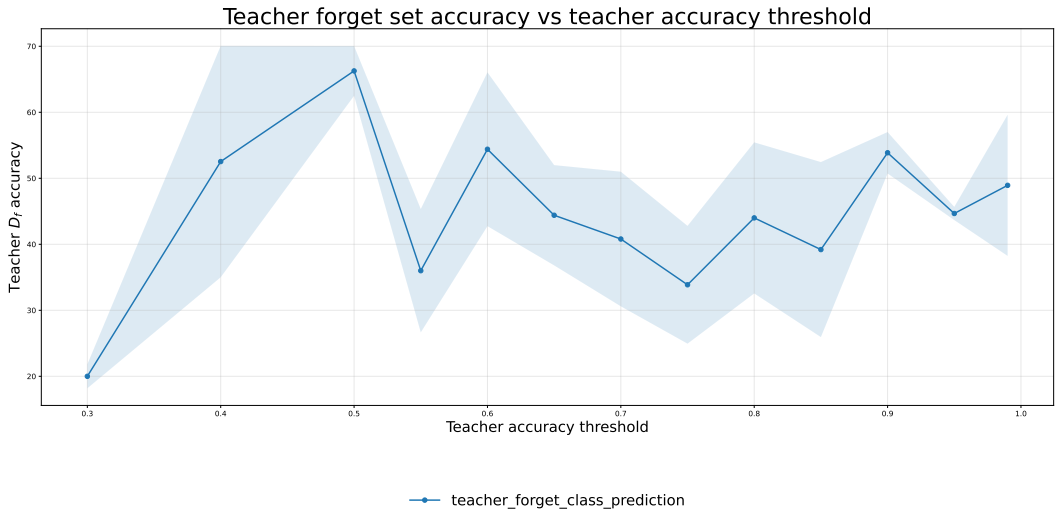

Figure 7: Teacher accuracy on the forget set (D_f) as a function of the training threshold. The teacher increasingly matches the retrained model as the threshold grows.

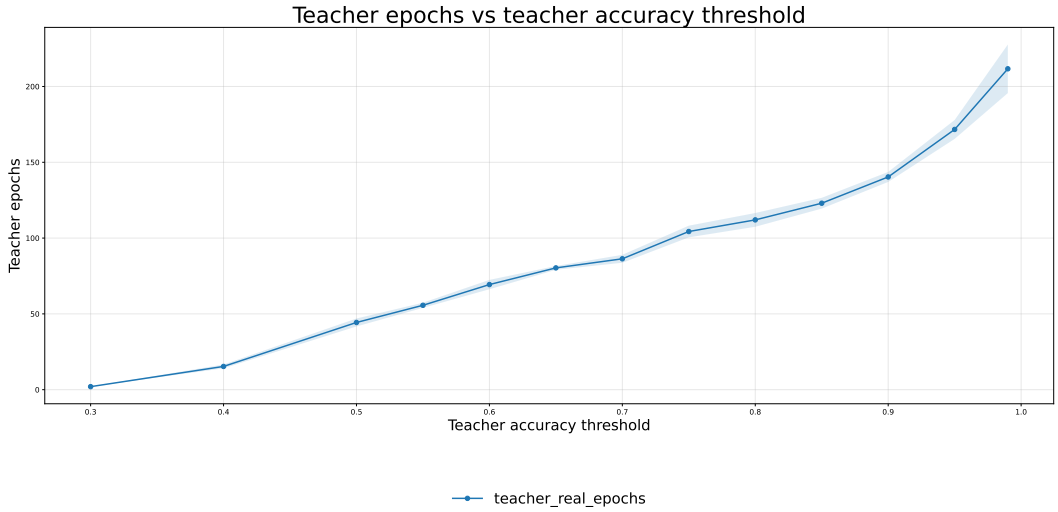

Figure 8: Number of training epochs required to reach the target teacher accuracy threshold on the support set. Higher thresholds lead to substantially increased training cost.

We vary the target teacher accuracy threshold on the support set in the range $[0.3, 0.99]$. Across this range, the final unlearning metrics remain largely stable, suggesting that the method is not highly sensitive to the exact stopping threshold once the teacher reaches moderate accuracy.

Overall, these results suggest that a moderately trained teacher is sufficient: increasing the support-set accuracy threshold within the considered range does not lead to substantial improvements in either aggregate or affected-class metrics.

D.4 Number of epochs

We study the effect of the number of unlearning epochs on performance. Figure 9 shows the evolution of affected-class metrics over training.

We observe that the model quickly approaches retraining behavior within the first few epochs. In particular, both retain and test accuracy stabilize early, while the forget accuracy rapidly decreases and then remains stable.

Overall, these results indicate that only a small number of unlearning steps is sufficient to capture the relevant local structure, and further training yields diminishing returns.

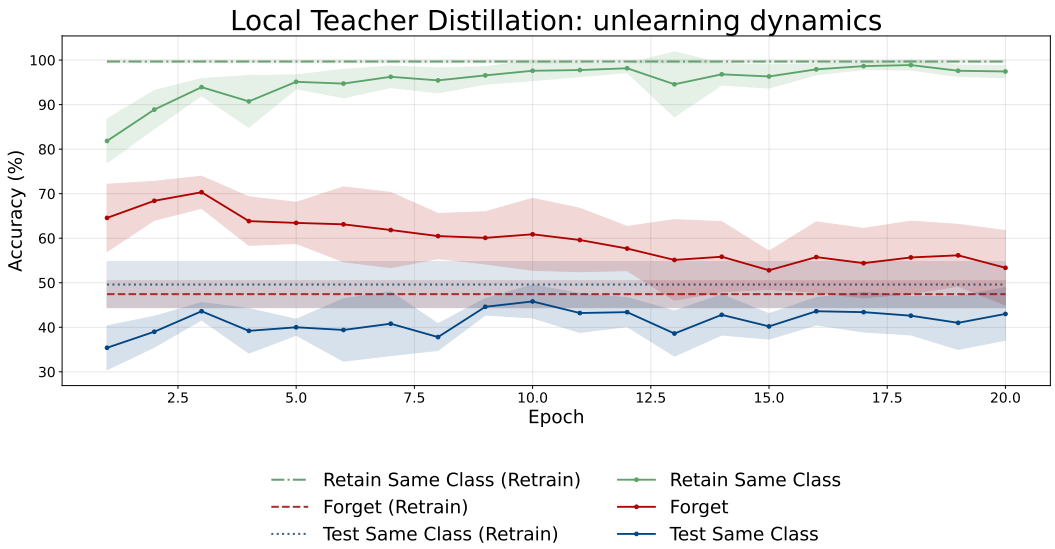

Figure 9: Unlearning dynamics as a function of the number of epochs. Solid lines show the proposed method, dashed lines indicate retraining performance. Metrics quickly stabilize, suggesting fast convergence to retraining behavior.

D.5 Support Size

We study the effect of the support size k on unlearning quality. Our goal is to identify a locally sufficient support set that approximates retraining without using the retrained model for selection. Because the final unlearning method uses top-3-renormalized teacher targets, all model predictions (teachers' and retrains') in this ablation are processed in the same way.

D.5.1 Retrain-free diagnostics

Using too small support sets leads to degenerate behavior: for instance, if the support set contains only points from the same class, the teacher trivially predicts that class and fails to capture the retrained decision boundary.

To analyze this effect, we vary k from 100 to 2700 (with the same step) and track several signals.

First, we measure the change in the predicted label distribution on D_f as the support size increases. Figure 10 shows the KL divergence between consecutive histogram predictions. We observe a transition point: for small k, the distribution changes smoothly, while beyond a certain point it starts to shift significantly at each step, indicating that the support set begins to incorporate noisy or less relevant points.

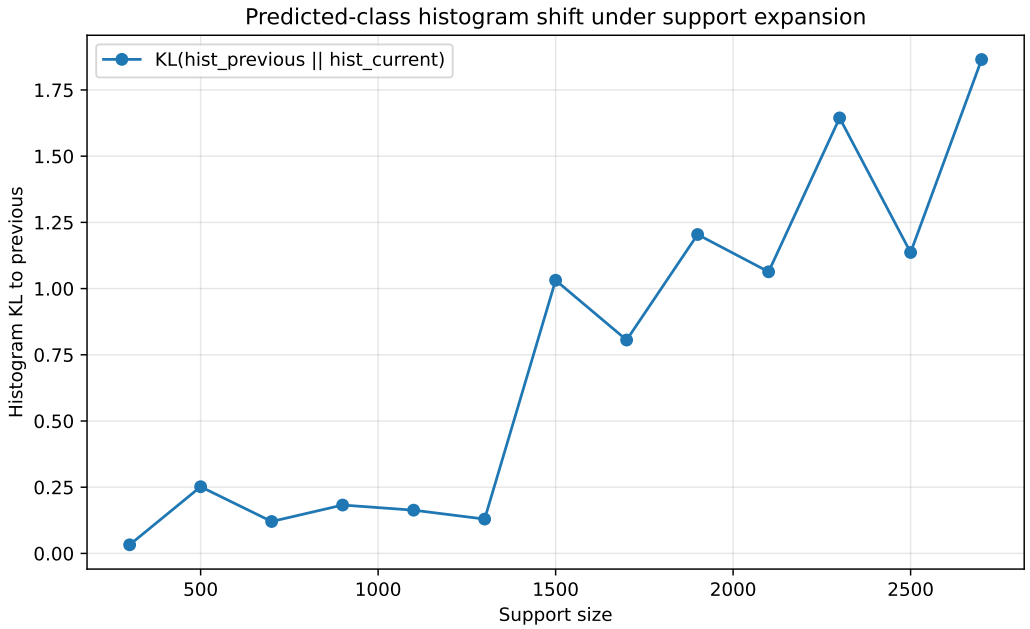

Figure 10: KL divergence between consecutive predicted-class histograms on D_f as the support size increases. A sharp increase indicates a transition from stable local structure to a noisier regime.

Second, we consider a retrain-free proxy: the accuracy on D_f. As shown in Figure 11, the accuracy initially decreases, then stabilizes, and finally begins to decrease again after the transition point, aligning with the onset of large distributional shifts.

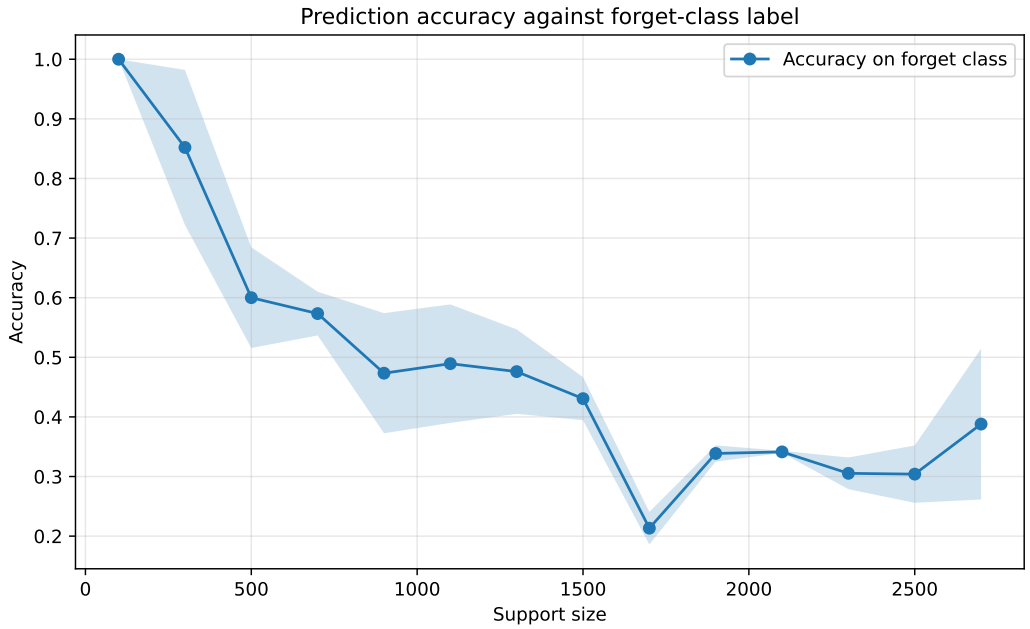

Figure 11: Accuracy on the forget class as a function of support size. The curve exhibits an initial decrease, followed by a plateau and a subsequent degradation, indicating the transition from sufficient to overly large support sets.

Finally, we relate this transition to similarity structure. The plateau region corresponds to similarity thresholds around 0.72–0.76, which matches the regime where local neighborhoods remain coherent. As a practical heuristic, we perform a single step of gradient ascent and measure accuracy drop across similarity bins (Figure 12). Points in high-similarity bins exhibit disproportionate degradation, suggesting that this signal can guide the choice of k.

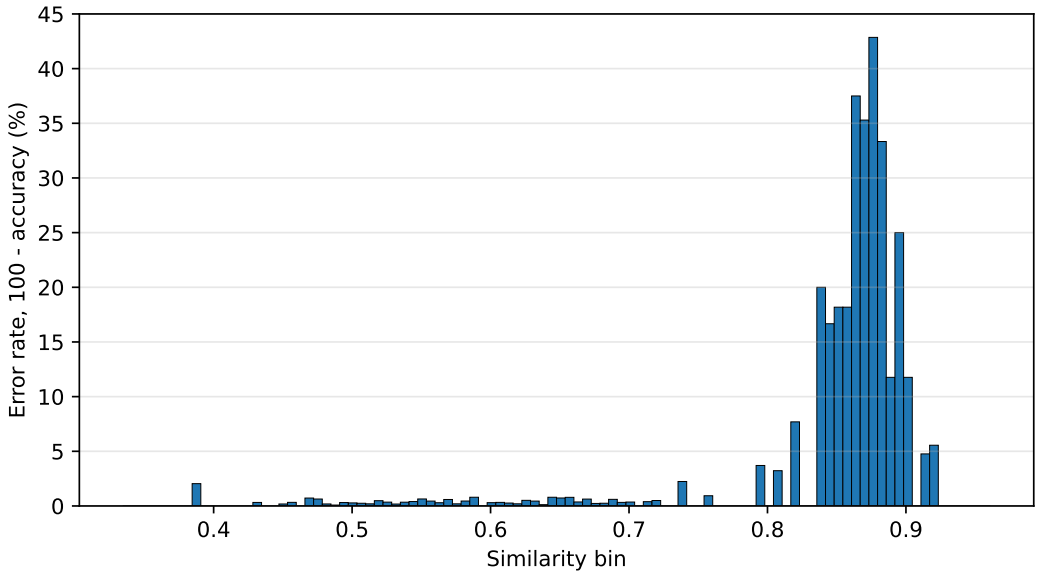

Figure 12: Accuracy drop across similarity bins after a single gradient ascent step. High-similarity points are disproportionately affected, providing a signal for selecting an appropriate support size.

Overall, these results indicate that there exists a regime of *locally sufficient* support sizes: beyond this regime, increasing k introduces noise and degrades the ability to capture the local structure relevant for unlearning.

D.5.2 Post-hoc comparison to retraining

We additionally compare the behavior of our method to retraining across several signals.

As a post-hoc validation, we compare the predicted-class distribution on D_f to that of retraining. Figure 13 shows that the KL divergence drops sharply once the support set becomes sufficiently large, and then changes only mildly. This supports the retrain-free diagnostics above: beyond a moderate local support size, adding more support examples yields limited additional alignment with retraining.

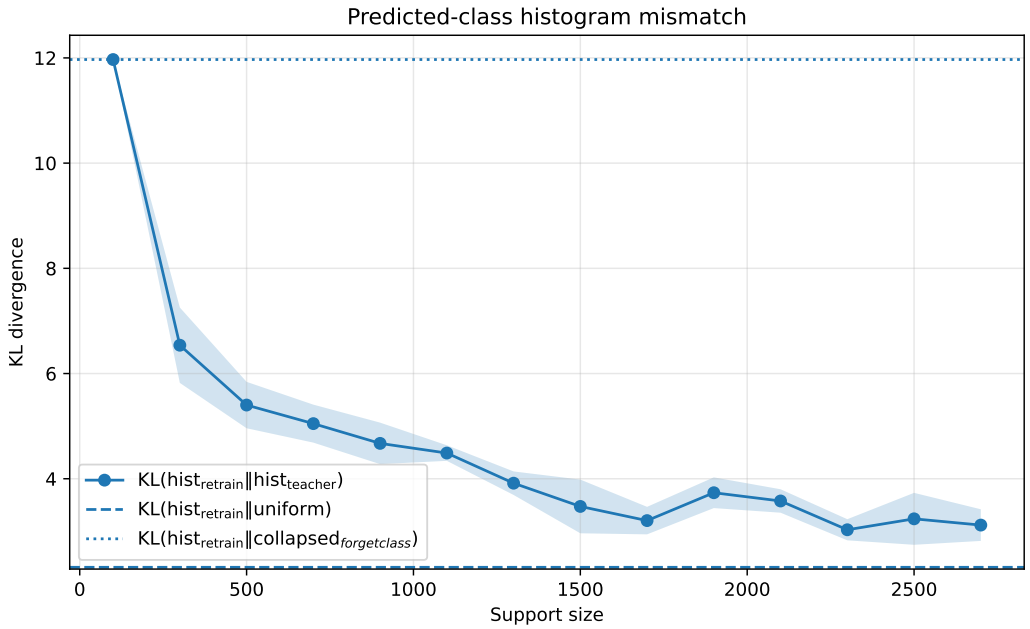

Figure 13: Post-hoc KL divergence between teacher and retrained predicted-class distributions on D_f as a function of support size. The divergence drops sharply for small support sizes and then changes only mildly, indicating that a moderate local support is sufficient for distribution-level alignment with retraining.

Together, these results confirm that increasing the support size beyond the locally sufficient regime does not improve approximation to retraining, and may even introduce additional noise.

E Additional Experiments

E.1 90% Class Deletion Experiments

We additionally evaluate our method in a more challenging setting, where 90% of a class is removed. This regime leaves only 10% of the affected-class training points in D_r, so retraining must preserve high accuracy on these remaining points while substantially reducing accuracy on D_f. This creates a large retain–forget prediction gap and makes approximation to retraining more difficult.

Aggregate and affected-class performance. Across aggregate metrics (Table 10), our method achieves the lowest Avg. Gap, indicating the closest overall approximation to retraining in this more challenging regime. Although LTD does not attain the smallest UA deviation, it provides the best trade-off across UA, RA, TA, and MIA.

Table 10: Main results for the 90% class deletion setting. Values are mean $\pm$ std over 5 runs; all accuracy metrics are in $\%$. Blue values show deviation from Retrain. Avg. Gap is the mean absolute deviation from Retrain over UA, RA, TA, and MIA.

Method	UA	RA	TA	MIA	Avg. Gap	RTE
Retrain	17.2 ± 3.0	99.9 ± 0.1	71.7 ± 0.3	96.1 ± 1.4	0	1564.7 ± 10.4
RL	46.4 ± 5.7 (+29.3)	94.3 ± 0.7 (-5.6)	67.5 ± 0.3 (-4.3)	100.0 ± 0.1 (+3.8)	10.7	162.0 ± 1.1
FT	60.2 ± 7.1 (+43.0)	94.0 ± 1.4 (-5.8)	67.6 ± 1.0 (-4.1)	50.4 ± 7.7 (-45.8)	24.7	155.2 ± 0.4
GA	24.8 ± 0.3 (+7.6)	94.8 ± 0.0 (-5.1)	66.2 ± 0.0 (-5.6)	83.9 ± 0.1 (-12.3)	7.6	1.9 ± 0.0
IU	19.2 ± 11.6 **(+2.0)**	97.8 ± 1.1 (-2.1)	69.1 ± 1.0 (-2.7)	91.3 ± 5.4 (-4.8)	2.9	27.3 ± 0.5
SalUn	40.0 ± 10.5 (+22.9)	93.6 ± 1.0 (-6.2)	67.0 ± 0.9 (-4.8)	100.0 ± 0.0 (+3.9)	9.4	361.1 ± 3.3
AMUN	32.5 ± 3.3 (+15.4)	94.3 ± 0.1 (-5.5)	72.5 ± 0.2 **(+0.8)**	81.2 ± 2.4 (-14.9)	9.2	173.1 ± 0.8
LTD (Ours)	11.4 ± 2.1 (-5.8)	99.4 ± 0.0 **(-0.4)**	70.4 ± 0.1 (-1.4)	99.2 ± 0.3 **(+3.0)**	2.7	215.6 ± 1.5

As in the 50% setting, aggregate metrics alone can be misleading. Table 11 shows that competing methods exhibit substantial degradation on affected-class points. In particular, methods such as IU and GA significantly deviate from retraining, despite relatively strong performance on some global metrics.

Table 11: Results on the affected-class subset for the 90% deletion setting. Values are mean ± std over 5 runs, in %. Blue values show deviation from Retrain. Avg. Gap is computed over $\text{RA}_{\text{affected-class}}$, $\text{UA}_{\text{affected-class}}$, and $\text{TA}_{\text{affected-class}}$.

Method	$\text{RA}_{\text{affected-class}}$	$\text{UA}_{\text{affected-class}}$	$\text{TA}_{\text{affected-class}}$	Avg. Gap
Retrain	98.8 ± 1.6	17.2 ± 3.0	17.8 ± 2.0	0
RL	68.0 ± 8.5 (-30.8)	46.4 ± 5.7 (+29.3)	26.0 ± 4.5 (+8.2)	22.8
FT	92.0 ± 2.8 **(-6.8)**	60.2 ± 7.1 (+43.0)	39.6 ± 4.2 (+21.8)	23.9
GA	32.4 ± 1.5 (-66.4)	24.8 ± 0.3 (+7.6)	16.0 ± 0.0 **(-1.8)**	25.3
IU	17.6 ± 14.0 (-81.2)	19.2 ± 11.6 **(+2.0)**	10.8 ± 5.3 (-7.0)	30.1
SalUn	58.8 ± 14.0 (-40.0)	40.0 ± 10.5 (+22.9)	24.8 ± 6.7 (+7.0)	23.3
AMUN	60.0 ± 7.0 (-38.8)	32.5 ± 3.3 (+15.4)	25.0 ± 1.1 (+7.2)	20.5
LTD (Ours)	75.6 ± 10.1 (-23.2)	11.4 ± 2.1 (-5.8)	9.4 ± 3.9 (-8.4)	12.5

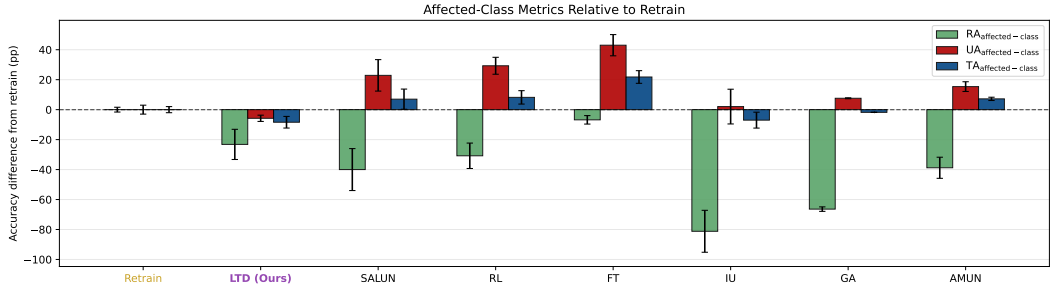

Figure 14: **Affected-class performance difference from Retrain, 90% of class deletion.** Bars show differences from Retrain for $\text{RA}_{\text{affected-class}}$, $\text{UA}_{\text{affected-class}}$, and $\text{TA}_{\text{affected-class}}$; values closer to zero indicate better agreement. The figure reveals localized collateral forgetting not captured by aggregate metrics.

In contrast, our method remains consistently closer to retraining across both aggregate and affected-class metrics. Notably, it achieves the lowest Avg. Gap on the affected class, indicating more effective removal of the target data, while maintaining reasonable performance on the remaining points.

Overall, these results confirm that the advantages of our approach persist in more extreme deletion regimes, where accurate control of local behavior becomes even more critical.

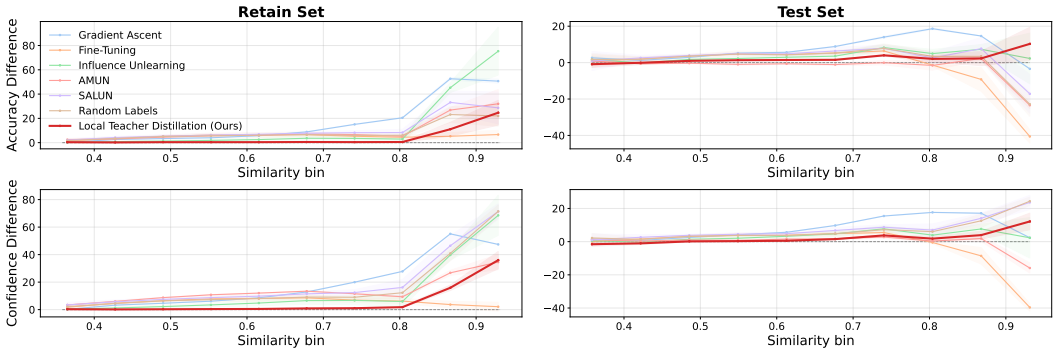

Figure 15: **Localized prediction deviations under 90% class deletion.** We plot bin-level differences from retraining: $\text{Acc}_{\text{retrain}} - \text{Acc}_{\text{unlearn}}$ for accuracy (top row) and $f_{\theta_{\text{retrain}}}(x)_y - f_{\theta_{\text{unlearn}}}(x)_y$ for correct-class confidence (bottom row), as functions of similarity to D_f on both retain and test sets. Larger deviations at high similarity indicate that unlearning effects concentrate near the removed data.

Similarity-based analysis. We further analyze locality effects in the 90% deletion setting by measuring accuracy and confidence differences with respect to retraining as a function of similarity to the forget set (Figure 15). We observe that deviations from retraining become significantly more pronounced for high-similarity points compared to the 50% setting. In particular, competing methods exhibit sharp degradation in both accuracy and confidence on points close to the forget set, indicating amplified localized collateral forgetting. In contrast, our method remains comparatively stable across similarity bins, maintaining low deviation from retraining even in this more challenging regime.

E.2 Additional affected classes under 50% deletion

Setup. We report additional results for the 50% partial-deletion setting on five randomly selected affected classes: `oak_tree` (52), `apple` (0), `hamster` (36), `road` (68), and `shark` (73). These experiments complement the main `couch` results and test whether localized collateral forgetting and the retraining alignment of LTD persist across different semantic classes. For each class, we report the same two views as in the main experiments: aggregate metrics and affected-class metrics. The aggregate table measures global agreement with retraining across UA, RA, TA, and MIA, while the affected-class table isolates the local neighborhood where collateral forgetting is most likely to appear.

E.2.1 Class 52: `oak_tree`

For `oak_tree`, LTD achieves the lowest aggregate Avg. Gap and the lowest affected-class Avg. Gap. This indicates that the method improves alignment with retraining both globally and on the localized affected-class subset. The improvement is not due to matching a single metric: LTD stays close to retraining on RA and TA, while also keeping the affected-class discrepancy moderate. Several baselines match retraining well on individual metrics, but their deviations are less balanced across the aggregate and affected-class views.

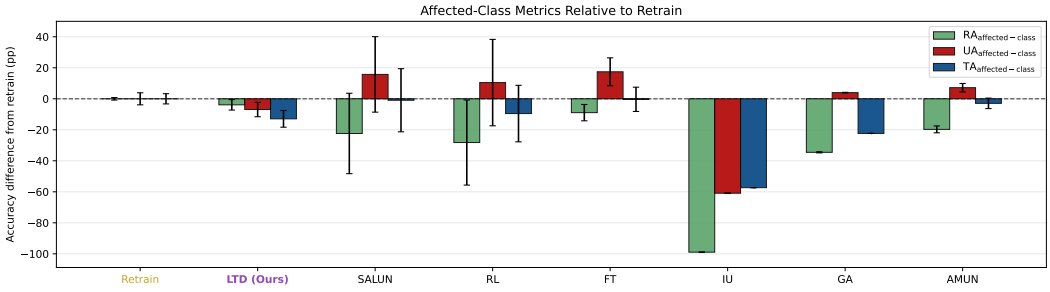

Figure 16: **Affected-class performance difference from Retrain, `oak_tree`.** Bars show differences from Retrain for $\text{RA}_{\text{affected-class}}$, $\text{UA}_{\text{affected-class}}$, and $\text{TA}_{\text{affected-class}}$; values closer to zero indicate better agreement. The figure reveals localized collateral forgetting not captured by aggregate metrics.

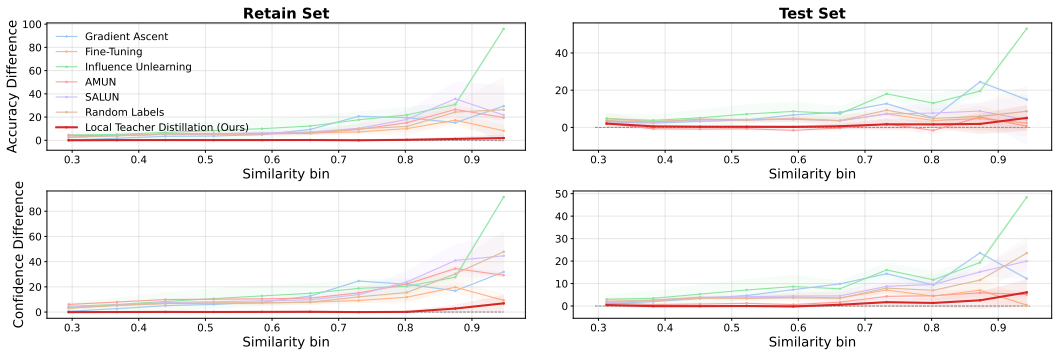

Figure 17: **Localized prediction deviations.** We plot bin-level differences from retraining: $\mathrm{Acc}_{\mathrm{retrain}} - \mathrm{Acc}_{\mathrm{unlearn}}$ for accuracy (top row) and $f_{\theta_{\mathrm{retrain}}}(x)_y - f_{\theta_{\mathrm{unlearn}}}(x)_y$ for correct-class confidence (bottom row), as functions of similarity to D_f on both retain and test sets. Larger deviations at high similarity indicate that unlearning effects concentrate near the removed data.

Table 12: Main results for the 50% deletion setting for class 52 (`oak_tree`). Values are mean ± std over 5 runs; all accuracy metrics are in %. Blue values show deviation from Retrain. Avg. Gap is the mean absolute deviation from Retrain over UA, RA, TA, and MIA.

Method	UA	RA	TA	MIA	Avg. Gap	RTE
Retrain	61.1 ± 3.9	99.9 ± 0.0	71.9 ± 0.3	64.1 ± 1.5	0	1587.9 ± 12.5
RL	71.5 ± 27.9 (+10.5)	94.2 ± 0.8 (-5.7)	67.7 ± 0.4 (-4.2)	89.4 ± 10.8 (+25.2)	11.4	157.3 ± 0.7
FT	78.5 ± 9.0 (+17.4)	94.4 ± 1.2 (-5.5)	67.8 ± 1.0 (-4.1)	32.6 ± 11.5 (-31.6)	14.6	156.5 ± 1.8
GA	65.0 ± 0.2 **(+4.0)**	95.5 ± 0.0 (-4.4)	67.3 ± 0.0 (-4.6)	49.4 ± 0.2 **(-14.8)**	6.9	1.7 ± 0.0
IU	0.2 ± 0.2 (-60.9)	91.2 ± 7.0 (-8.7)	64.9 ± 4.3 (-7.0)	100.0 ± 0.0 (+35.9)	28.1	27.7 ± 0.4
SalUn	76.8 ± 24.3 (+15.7)	94.0 ± 1.1 (-5.8)	67.9 ± 0.6 (-4.0)	82.1 ± 16.1 (+17.9)	10.9	350.2 ± 5.4
AMUN	68.2 ± 2.7 (+7.2)	94.4 ± 0.1 (-5.5)	72.7 ± 0.3 **(+0.8)**	49.3 ± 2.9 (-14.9)	7.1	151.2 ± 0.7
LTD (Ours)	54.2 ± 4.7 (-6.9)	99.6 ± 0.1 **(-0.3)**	71.1 ± 0.3 **(-0.8)**	80.9 ± 3.1 (+16.7)	6.2	215.4 ± 2.9

Table 13: Results on the affected-class subset for the 50% deletion setting for class 52 (`oak_tree`). Values are mean ± std over 5 runs, in %. Blue values show deviation from Retrain. Avg. Gap is computed over $\mathrm{RA}_{\text{affected-class}}$, $\mathrm{UA}_{\text{affected-class}}$, and $\mathrm{TA}_{\text{affected-class}}$.

Method	$\mathrm{RA}_{\text{affected-class}}$	$\mathrm{UA}_{\text{affected-class}}$	$\mathrm{TA}_{\text{affected-class}}$	Avg. Gap
Retrain	98.9 ± 0.8	61.1 ± 3.9	57.3 ± 3.3	0
RL	70.7 ± 27.4 (-28.2)	71.5 ± 27.9 (+10.5)	47.8 ± 18.2 (-9.5)	16.1
FT	90.0 ± 5.3 (-8.9)	78.5 ± 9.0 (+17.4)	57.0 ± 7.8 **(-0.3)**	8.9
GA	64.4 ± 0.4 (-34.5)	65.0 ± 0.2 **(+4.0)**	35.0 ± 0.0 (-22.3)	20.3
IU	0.0 ± 0.0 (-98.9)	0.2 ± 0.2 (-60.9)	0.0 ± 0.0 (-57.3)	72.4
SalUn	76.6 ± 25.9 (-22.4)	76.8 ± 24.3 (+15.7)	56.4 ± 20.4 (-0.9)	13.0
AMUN	79.2 ± 2.2 (-19.7)	68.2 ± 2.7 (+7.2)	54.4 ± 3.4 (-2.9)	9.9
LTD (Ours)	95.0 ± 3.4 **(-3.9)**	54.2 ± 4.7 (-6.9)	44.4 ± 5.4 (-12.9)	7.9

E.2.2 Class 0: `apple`

For `apple`, LTD obtains the best aggregate Avg. Gap, showing the closest overall agreement with retraining across UA, RA, TA, and MIA. On the affected-class subset, however, AMUN achieves the best Avg. Gap, with LTD ranking second. This is a useful case where another method matches the local affected-class behavior slightly better, while LTD remains more balanced globally. The comparison highlights that the aggregate and affected-class views are complementary: a method can be strong locally but less aligned overall, and vice versa.

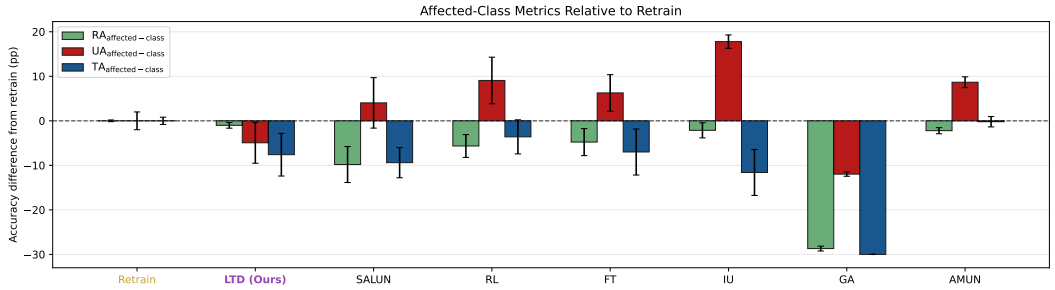

Figure 18: **Affected-class performance difference from Retrain.** Bars show differences from Retrain for $\text{RA}_{\text{affected-class}}$, $\text{UA}_{\text{affected-class}}$, and $\text{TA}_{\text{affected-class}}$; values closer to zero indicate better agreement. The figure reveals localized collateral forgetting not captured by aggregate metrics.

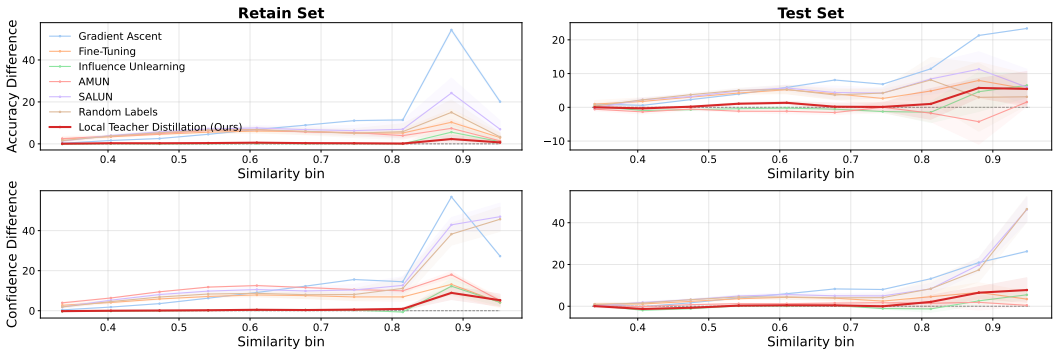

Figure 19: **Localized prediction deviations.** We plot bin-level differences from retraining: $\text{Acc}_{\text{retrain}} - \text{Acc}_{\text{unlearn}}$ for accuracy (top row) and $f_{\theta_{\text{retrain}}}(x)_y - f_{\theta_{\text{unlearn}}}(x)_y$ for correct-class confidence (bottom row), as functions of similarity to D_f on both retain and test sets. Larger deviations at high similarity indicate that unlearning effects concentrate near the removed data.

Table 14: Main results for the 50% deletion setting for class 0 (`apple`). Values are mean ± std over 5 runs; all accuracy metrics are in %. Blue values show deviation from Retrain. Avg. Gap is the mean absolute deviation from Retrain over UA, RA, TA, and MIA.

Method	UA	RA	TA	MIA	Avg. Gap	RTE
Retrain	79.3 ± 2.0	99.8 ± 0.0	71.7 ± 0.1	35.7 ± 1.8	0	1540.6 ± 10.1
RL	88.4 ± 5.2 (+9.1)	94.1 ± 0.9 (-5.7)	67.5 ± 0.6 (-4.3)	94.4 ± 3.2 (+58.7)	19.4	158.0 ± 0.9
FT	85.6 ± 4.1 (+6.3)	94.8 ± 0.7 (-5.0)	68.0 ± 0.6 (-3.7)	21.2 ± 5.3 (-14.5)	7.4	155.7 ± 1.0
GA	67.4 ± 0.5 (-12.0)	95.5 ± 0.0 (-4.3)	67.8 ± 0.0 (-3.9)	52.2 ± 0.2 (+16.5)	9.2	1.7 ± 0.0
IU	97.1 ± 1.5 (+17.8)	99.8 ± 0.1 **(-0.1)**	72.2 ± 0.3 **(+0.5)**	25.2 ± 5.5 **(-10.5)**	7.2	27.5 ± 0.3
SalUn	83.4 ± 5.7 **(+4.0)**	93.6 ± 0.6 (-6.2)	67.5 ± 0.3 (-4.2)	90.5 ± 5.0 (+54.7)	17.3	351.1 ± 2.4
AMUN	88.0 ± 1.2 (+8.7)	94.3 ± 0.1 (-5.5)	72.7 ± 0.3 (+1.0)	17.8 ± 1.6 (-18.0)	8.3	171.7 ± 1.0
LTD (Ours)	74.4 ± 4.6 (-4.9)	99.5 ± 0.0 (-0.4)	71.0 ± 0.1 (-0.7)	49.8 ± 10.6 (+14.1)	5.0	213.7 ± 1.5

Table 15: Results on the affected-class subset for the 50% deletion setting for class 0 (`apple`). Values are mean ± std over 5 runs, in %. Blue values show deviation from Retrain. Avg. Gap is computed over $\text{RA}_{\text{affected-class}}$, $\text{UA}_{\text{affected-class}}$, and $\text{TA}_{\text{affected-class}}$.

Method	$\text{RA}_{\text{affected-class}}$	$\text{UA}_{\text{affected-class}}$	$\text{TA}_{\text{affected-class}}$	Avg. Gap
Retrain	99.7 ± 0.2	79.3 ± 2.0	82.0 ± 0.8	0
RL	94.1 ± 2.6 (-5.7)	88.4 ± 5.2 (+9.1)	78.4 ± 3.8 (-3.6)	6.1
FT	95.0 ± 3.0 (-4.8)	85.6 ± 4.1 (+6.3)	75.0 ± 5.2 (-7.0)	6.0
GA	71.0 ± 0.5 (-28.7)	67.4 ± 0.5 (-12.0)	52.0 ± 0.0 (-30.0)	23.6
IU	97.6 ± 1.7 (-2.1)	97.1 ± 1.5 (+17.8)	70.4 ± 5.2 (-11.6)	10.5
SalUn	89.9 ± 4.0 (-9.8)	83.4 ± 5.7 (+4.0)	72.6 ± 3.4 (-9.4)	7.7
AMUN	97.5 ± 0.7 (-2.2)	88.0 ± 1.2 (+8.7)	81.8 ± 1.2 (-0.2)	3.7
LTD (Ours)	98.7 ± 0.6 (-1.0)	74.4 ± 4.6 (-4.9)	74.4 ± 4.8 (-7.6)	4.5

E.2.3 Class 36: `hamster`

For `hamster`, LTD achieves the lowest aggregate Avg. Gap and the lowest affected-class Avg. Gap. The method stays close to retraining on RA, TA, and MIA, while also reducing the local affected-class discrepancy relative to the baselines. This class is therefore consistent with the broader trend across additional affected classes: LTD provides balanced agreement with retraining both globally and in the neighborhood most directly coupled to the forget set.

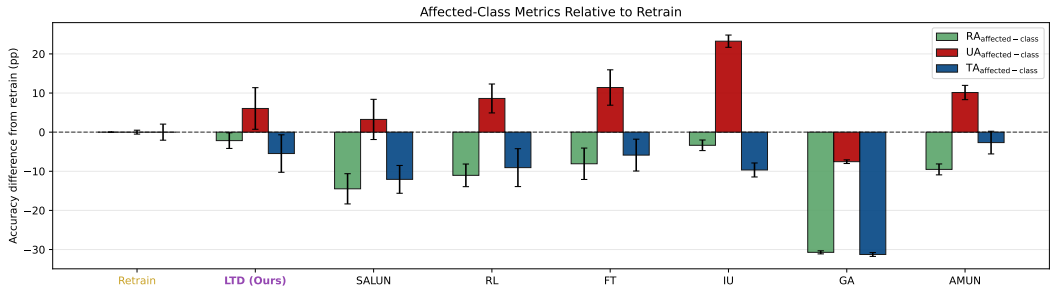

Figure 20: **Affected-class performance difference from Retrain, `hamster` (36).** Bars show differences from Retrain for $\text{RA}_{\text{affected-class}}$, $\text{UA}_{\text{affected-class}}$, and $\text{TA}_{\text{affected-class}}$; values closer to zero indicate better agreement. The figure reveals localized collateral forgetting not captured by aggregate metrics.

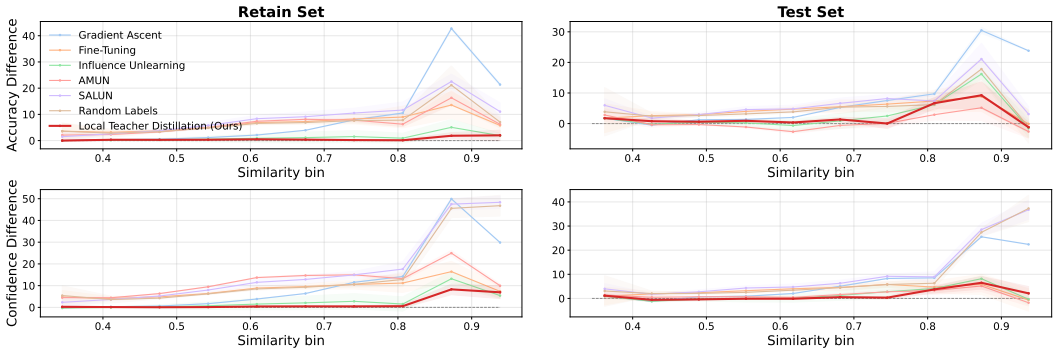

Figure 21: **Localized prediction deviations, `hamster` (36).** We plot bin-level differences from retraining: $\text{Acc}_{\text{retrain}} - \text{Acc}_{\text{unlearn}}$ for accuracy (top row) and $f_{\theta_{\text{retrain}}}(x)_y - f_{\theta_{\text{unlearn}}}(x)_y$ for correct-class confidence (bottom row), as functions of similarity to D_f on both retain and test sets. Larger deviations at high similarity indicate that unlearning effects concentrate near the removed data.

Table 16: Main results for the 50% deletion setting for class 36 (`hamster`). Values are mean ± std over 5 runs; all accuracy metrics are in %. Blue values show deviation from Retrain. Avg. Gap is the mean absolute deviation from Retrain over UA, RA, TA, and MIA.

Method	UA	RA	TA	MIA	Avg. Gap	RTE
Retrain	72.7 ± 0.5	99.9 ± 0.0	71.8 ± 0.4	47.3 ± 0.4	0	1580.3 ± 3.9
RL	81.3 ± 3.7 (+8.6)	94.6 ± 0.3 (-5.2)	68.2 ± 0.2 (-3.6)	96.4 ± 2.4 (+49.1)	16.6	157.4 ± 1.0
FT	84.1 ± 4.5 (+11.4)	94.4 ± 0.5 (-5.5)	67.7 ± 0.5 (-4.1)	23.4 ± 5.3 (-24.0)	11.3	155.9 ± 1.3
GA	65.1 ± 0.5 (-7.5)	97.9 ± 0.0 (-1.9)	69.4 ± 0.0 (-2.4)	64.4 ± 0.0 +17.1	7.2	1.7 ± 0.0
IU	95.9 ± 1.6 (+23.3)	99.2 ± 1.2 (-0.7)	71.4 ± 1.3 **(-0.3)**	28.1 ± 5.6 (-19.3)	10.9	27.3 ± 0.4
SalUn	75.9 ± 5.1 **(+3.3)**	93.4 ± 1.0 (-6.4)	67.2 ± 0.9 (-4.6)	92.0 ± 5.7 (+44.7)	14.8	352.5 ± 2.8
AMUN	82.8 ± 1.8 (+10.1)	94.3 ± 0.2 (-5.5)	72.9 ± 0.1 (+1.1)	26.5 ± 3.6 (-20.9)	9.4	169.9 ± 1.3
LTD (Ours)	78.7 ± 5.3 (+6.1)	99.5 ± 0.1 **(-0.4)**	71.0 ± 0.3 (-0.8)	48.1 ± 9.4 **(+0.7)**	2.0	214.8 ± 3.8

Table 17: Results on the affected-class subset for the 50% deletion setting for class 36 (`hamster`). Values are mean ± std over 5 runs, in %. Blue values show deviation from Retrain. Avg. Gap is computed over $\text{RA}_{\text{affected-class}}$, $\text{UA}_{\text{affected-class}}$, and $\text{TA}_{\text{affected-class}}$.

Method	$\text{RA}_{\text{affected-class}}$	$\text{UA}_{\text{affected-class}}$	$\text{TA}_{\text{affected-class}}$	Avg. Gap
Retrain	100.0 ± 0.0	72.7 ± 0.5	72.7 ± 2.1	0
RL	89.0 ± 2.9 (-11.0)	81.3 ± 3.7 (+8.6)	63.6 ± 4.8 (-9.1)	9.6
FT	91.9 ± 4.0 (-8.1)	84.1 ± 4.5 (+11.4)	66.8 ± 4.1 (-5.9)	8.5
GA	69.3 ± 0.4 (-30.7)	65.1 ± 0.5 (-7.5)	41.4 ± 0.5 (-31.3)	23.2
IU	96.6 ± 1.4 -3.4	95.9 ± 1.6 (+23.3)	63.0 ± 1.8 (-9.7)	12.1
SalUn	85.5 ± 3.9 (-14.5)	75.9 ± 5.1 **(+3.3)**	60.6 ± 3.6 (-12.1)	9.9
AMUN	90.5 ± 1.4 (-9.5)	82.8 ± 1.8 (+10.1)	70.0 ± 2.9 **(-2.7)**	7.4
LTD (Ours)	97.8 ± 2.0 **(-2.2)**	78.7 ± 5.3 (+6.1)	67.2 ± 4.8 (-5.5)	4.6

E.2.4 Class 68: `road`

For `road`, LTD achieves the lowest aggregate Avg. Gap. On the affected-class subset, FT, LTD, and AMUN are very close, with LTD ranking second and nearly matching the best affected-class Avg. Gap. This class shows a regime where several methods can approximate the local affected-class behavior reasonably well, but LTD gives the best overall trade-off across the aggregate metrics. In particular, it preserves RA and TA close to retraining while keeping the MIA deviation smaller than most baselines.

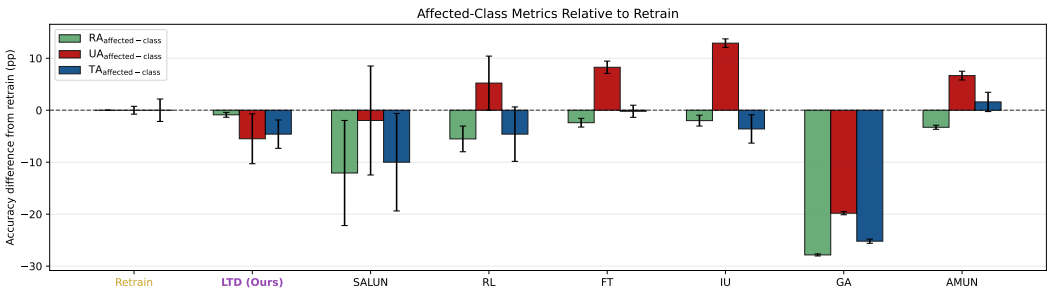

Figure 22: **Affected-class performance difference from Retrain, `road` (68).** Bars show differences from Retrain for $\text{RA}_{\text{affected-class}}$, $\text{UA}_{\text{affected-class}}$, and $\text{TA}_{\text{affected-class}}$; values closer to zero indicate better agreement. The figure reveals localized collateral forgetting not captured by aggregate metrics.

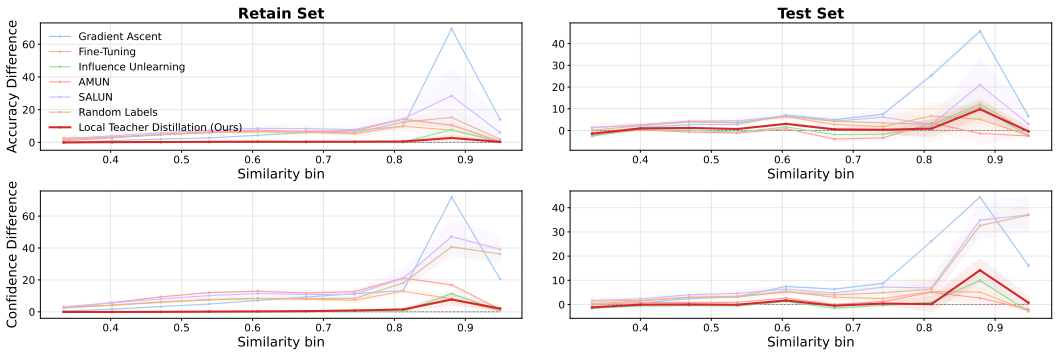

Figure 23: **Localized prediction deviations, `road` (68).** We plot bin-level differences from retraining: $\mathrm{Acc}_{\mathrm{retrain}} - \mathrm{Acc}_{\mathrm{unlearn}}$ for accuracy (top row) and $f_{\theta_{\mathrm{retrain}}}(x)_y - f_{\theta_{\mathrm{unlearn}}}(x)_y$ for correct-class confidence (bottom row), as functions of similarity to D_f on both retain and test sets. Larger deviations at high similarity indicate that unlearning effects concentrate near the removed data.

Table 18: Main results for the 50% deletion setting for class 68 (`road`). Values are mean ± std over 5 runs; all accuracy metrics are in %. Blue values show deviation from Retrain. Avg. Gap is the mean absolute deviation from Retrain over UA, RA, TA, and MIA.

Method	UA	RA	TA	MIA	Avg. Gap	RTE
Retrain	86.1 ± 0.8	99.8 ± 0.1	72.0 ± 0.3	26.3 ± 2.5	0	1557.8 ± 12.1
RL	91.4 ± 5.2 (+5.2)	94.6 ± 0.7 (-5.3)	68.0 ± 0.4 (-4.0)	85.7 ± 6.4 (+59.4)	18.5	157.8 ± 0.6
FT	94.4 ± 1.2 (+8.3)	94.6 ± 0.5 (-5.2)	68.0 ± 0.4 (-4.0)	8.6 ± 2.6 (-17.7)	8.8	156.0 ± 0.6
GA	66.3 ± 0.3 (-19.8)	97.0 ± 0.0 (-2.8)	68.4 ± 0.0 (-3.7)	50.2 ± 0.3 (+24.0)	12.6	1.7 ± 0.0
IU	99.0 ± 0.8 (+12.9)	99.8 ± 0.1 **(-0.0)**	72.1 ± 0.3 **(+0.0)**	12.5 ± 3.0 (-13.8)	6.7	27.4 ± 0.3
SalUn	84.2 ± 10.5 **(-2.0)**	93.2 ± 1.2 (-6.6)	67.4 ± 0.9 (-4.6)	77.8 ± 14.8 (+51.5)	16.2	347.4 ± 3.9
AMUN	92.8 ± 0.8 (+6.7)	94.3 ± 0.1 (-5.5)	72.8 ± 0.3 (+0.7)	11.1 ± 1.2 (-15.1)	7.0	171.1 ± 0.5
LTD (Ours)	80.6 ± 4.8 (-5.5)	99.5 ± 0.0 (-0.3)	70.8 ± 0.2 (-1.2)	33.1 ± 8.1 **(+6.9)**	3.5	202.6 ± 3.4

Table 19: Results on the affected-class subset for the 50% deletion setting for class 68 (`road`). Values are mean ± std over 5 runs, in %. Blue values show deviation from Retrain. Avg. Gap is computed over $\mathrm{RA}_{\text{affected-class}}$, $\mathrm{UA}_{\text{affected-class}}$, and $\mathrm{TA}_{\text{affected-class}}$.

Method	$\mathrm{RA}_{\text{affected-class}}$	$\mathrm{UA}_{\text{affected-class}}$	$\mathrm{TA}_{\text{affected-class}}$	Avg. Gap
Retrain	100.0 ± 0.0	86.1 ± 0.8	88.0 ± 2.2	0
RL	94.5 ± 2.5 (-5.5)	91.4 ± 5.2 (+5.2)	83.4 ± 5.2 (-4.6)	5.1
FT	97.6 ± 0.8 (-2.4)	94.4 ± 1.2 (+8.3)	87.8 ± 1.2 **(-0.2)**	3.6
GA	72.2 ± 0.2 (-27.8)	66.3 ± 0.3 (-19.8)	62.8 ± 0.4 (-25.2)	24.3
IU	98.0 ± 1.0 (-2.0)	99.0 ± 0.8 (+12.9)	84.4 ± 2.7 (-3.6)	6.2
SalUn	87.9 ± 10.1 (-12.1)	84.2 ± 10.5 **(-2.0)**	78.0 ± 9.4 (-10.0)	8.0
AMUN	96.7 ± 0.4 (-3.3)	92.8 ± 0.8 (+6.7)	89.6 ± 1.9 (+1.6)	3.8
LTD (Ours)	99.1 ± 0.5 **(-0.9)**	80.6 ± 4.8 (-5.5)	83.4 ± 2.7 (-4.6)	3.7

E.2.5 Class 73: `shark`

For `shark`, LTD is the clearest positive case among the additional classes: it achieves the lowest aggregate Avg. Gap and the lowest affected-class Avg. Gap. The method stays close to retraining on RA, TA, and MIA, while also reducing the mismatch on the affected-class subset. Competing methods often improve one component at the cost of another, for example by preserving global utility but substantially shifting UA or degrading affected-class accuracy. This supports the interpretation that local teacher targets can better match the retraining-induced local prediction structure.

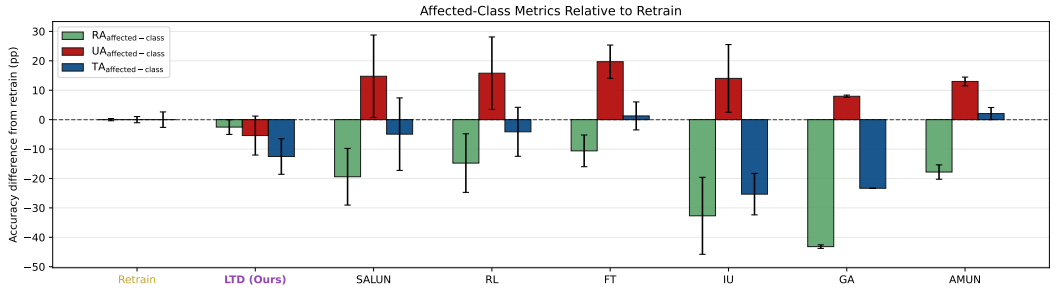

Figure 24: **Affected-class performance difference from Retrain.** Bars show differences from Retrain for $\text{RA}_{\text{affected-class}}$, $\text{UA}_{\text{affected-class}}$, and $\text{TA}_{\text{affected-class}}$; values closer to zero indicate better agreement. The figure reveals localized collateral forgetting not captured by aggregate metrics.

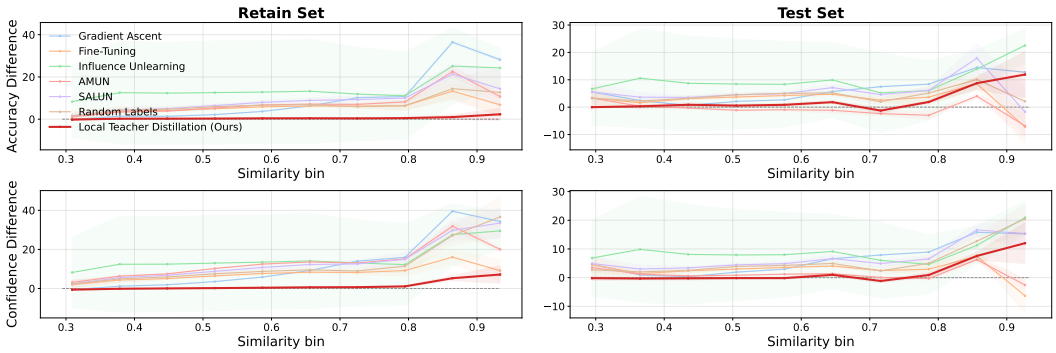

Figure 25: **Localized prediction deviations.** We plot bin-level differences from retraining: $\text{Acc}_{\text{retrain}} - \text{Acc}_{\text{unlearn}}$ for accuracy (top row) and $f_{\theta_{\text{retrain}}}(x)_y - f_{\theta_{\text{unlearn}}}(x)_y$ for correct-class confidence (bottom row), as functions of similarity to D_f on both retain and test sets. Larger deviations at high similarity indicate that unlearning effects concentrate near the removed data.

Table 20: Main results for the 50% deletion setting for class 73 (`shark`). Values are mean ± std over 5 runs; all accuracy metrics are in %. Blue values show deviation from Retrain. Avg. Gap is the mean absolute deviation from Retrain over UA, RA, TA, and MIA.

Method	UA	RA	TA	MIA	Avg. Gap	RTE
Retrain	54.9 ± 1.0	99.8 ± 0.0	71.8 ± 0.3	70.4 ± 0.9	0	1581.3 ± 4.6
RL	70.7 ± 12.3 (+15.8)	94.0 ± 0.9 (-5.8)	67.5 ± 0.5 (-4.3)	90.7 ± 5.5 (+20.3)	11.5	157.9 ± 0.8
FT	74.6 ± 5.7 (+19.7)	94.8 ± 0.3 (-5.1)	68.1 ± 0.3 (-3.7)	37.0 ± 8.4 (-33.4)	15.5	155.8 ± 0.8
GA	62.9 ± 0.4 (+7.9)	96.6 ± 0.0 (-3.3)	68.8 ± 0.0 (-3.0)	58.2 ± 0.6 (-12.2)	6.6	1.7 ± 0.0
IU	69.0 ± 11.5 (+14.0)	87.2 ± 24.0 (-12.7)	63.2 ± 15.7 (-8.6)	62.0 ± 19.2 (-8.4)	10.9	27.2 ± 0.3
SalUn	69.7 ± 14.0 (+14.7)	93.1 ± 0.7 (-6.7)	66.9 ± 0.4 (-4.9)	79.5 ± 6.4 (+9.1)	8.8	351.5 ± 1.7
AMUN	67.9 ± 1.5 (+13.0)	94.3 ± 0.0 (-5.5)	72.8 ± 0.2 (+1.0)	47.1 ± 1.3 (-23.3)	10.7	173.5 ± 0.9
LTD (Ours)	49.5 ± 6.6 (-5.4)	99.5 ± 0.1 (-0.3)	70.9 ± 0.4 (-0.9)	76.3 ± 5.9 (+5.9)	3.1	210.3 ± 0.7

Table 21: Results on the affected-class subset for the 50% deletion setting for class 73 (`shark`). Values are mean ± std over 5 runs, in %. Blue values show deviation from Retrain. Avg. Gap is computed over $RA_{affected\text{-}class}$, $UA_{affected\text{-}class}$, and $TA_{affected\text{-}class}$.

Method	$RA_{affected\text{-}class}$	$UA_{affected\text{-}class}$	$TA_{affected\text{-}class}$	Avg. Gap
Retrain	99.7 ± 0.4	54.9 ± 1.0	43.3 ± 2.6	0
RL	85.0 ± 10.0 (-14.8)	70.7 ± 12.3 (+15.8)	39.2 ± 8.3 (-4.1)	11.6
FT	89.1 ± 5.4 (-10.6)	74.6 ± 5.7 (+19.7)	44.6 ± 4.8 **(+1.3)**	10.5
GA	56.6 ± 0.6 (-43.2)	62.9 ± 0.4 (+7.9)	20.0 ± 0.0 (-23.3)	24.8
IU	67.0 ± 13.1 (-32.7)	69.0 ± 11.5 (+14.0)	18.0 ± 7.0 (-25.3)	24.0
SalUn	80.3 ± 9.6 (-19.4)	69.7 ± 14.0 (+14.7)	38.4 ± 12.3 (-4.9)	13.0
AMUN	81.9 ± 2.4 (-17.8)	67.9 ± 1.5 (+13.0)	45.4 ± 2.1 (+2.1)	11.0
LTD (Ours)	97.2 ± 2.5 **(-2.5)**	49.5 ± 6.6 **(-5.4)**	30.8 ± 6.0 (-12.5)	6.8

E.2.6 Summary across classes

Overall, the additional classes support the main conclusion that localized collateral forgetting is not specific to the `couch` class. Across all five randomly selected affected classes, LTD achieves the lowest aggregate Avg. Gap. This shows consistent global alignment with retraining across UA, RA, TA, and MIA, rather than a result driven by a favorable choice of the main affected class.

The affected-class results provide a more localized view. LTD achieves the best affected-class Avg. Gap on three of the five classes and ranks second on the remaining two. Thus, while the method is uniformly best in aggregate, the local affected-class comparison is more competitive. This distinction is important: aggregate metrics measure overall agreement with retraining, whereas affected-class metrics isolate the neighborhood where collateral forgetting is most likely to appear.

Importantly, the retraining target is class-dependent. In partial-class deletion, there is no single desired level of UA or affected-class test accuracy: classes such as `road` and `shark` induce different retraining trade-offs between forgetting D_f and preserving same-class test behavior. Thus, the setting requires more than uniformly suppressing the deleted examples; the unlearning method must flexibly match the class-specific retraining behavior. Notably, the LTD teacher is not merely preserving the original class identity: in partial deletion, the retrained model can assign nontrivial soft predictions to deleted examples, which are not determined by the original label alone.

AMUN is the strongest competing baseline in these additional experiments, especially on affected-class metrics. It achieves the best affected-class Avg. Gap for `apple` and remains close to the best methods for `road`. LTD is more consistent across the two evaluation views: it is best on all aggregate tables and remains among the top two on all affected-class tables. This suggests that local teacher targets improve retraining alignment systematically across classes, rather than only for a single deletion target or a single metric.

E.3 Additional Experiments on SVHN with ViT-Tiny

To assess the generalisability of the compared methods beyond the CIFAR setting, we conduct additional experiments on the SVHN dataset [20] with a ViT-Tiny backbone [7] (patch size 4, image size 32×32, 10 output classes) and apply class-forgetting for class 9. We consider two forgetting fractions: $\rho = 0.5$, where half of the class-9 training samples are designated as the forget set, and the more challenging $\rho = 0.9$. All methods are evaluated across 5 random seeds.

Tables 22 and 23 show the results for 50% deletion. LTD achieves the lowest global Avg. Gap to retraining, improving over the strongest baseline AMUN (2.6 vs. 4.0). The affected-class results show the same trend: LTD obtains the lowest affected-class Avg. Gap and closely matches the retrained model on affected-class test accuracy. The corresponding bar plot in Figure 26 shows a competitive affected-class comparison: LTD closely matches retraining on affected-class RA and TA, with small variance across runs, although it still overshoots UA.

Table 22: SVHN + ViT-Tiny results for partial deletion of digit 9 with class-fraction-to-forget $= 0.5$. Values are mean $\pm$ std over 5 runs; accuracy metrics are in % and RTE is in seconds. Blue values show deviation from Retrain. Avg. Gap is the mean absolute deviation from Retrain over UA, RA, TA, and MIA.

Method	UA	RA	TA	MIA	Avg. Gap	RTE
Retrain	74.9 ± 1.1	100.0 ± 0.0	84.9 ± 0.3	29.9 ± 1.6	0	2115.0 ± 5.8
RL	90.3 ± 0.9 (+15.4)	99.7 ± 0.0 (-0.3)	84.2 ± 0.0 (-0.6)	99.9 ± 0.1 (+70.0)	21.6	197.6 ± 0.9
FT	40.0 ± 48.9 (-34.9)	61.2 ± 32.9 (-38.8)	55.4 ± 25.9 (-29.5)	58.2 ± 47.6 (+28.3)	32.8	187.5 ± 1.2
GA	100.0 ± 0.0 (+25.1)	100.0 ± 0.0 **(+0.0)**	85.2 ± 0.0 (+0.3)	0.0 ± 0.0 (-29.9)	13.8	5.3 ± 0.1
IU	100.0 ± 0.0 (+25.1)	100.0 ± 0.0 (+0.0)	85.2 ± 0.0 **(+0.3)**	0.0 ± 0.0 (-29.9)	13.8	36.8 ± 0.2
SalUn	86.7 ± 1.1 (+11.8)	99.6 ± 0.0 (-0.4)	84.1 ± 0.1 (-0.8)	98.6 ± 0.6 (+68.7)	20.4	356.4 ± 1.1
AMUN	75.2 ± 14.0 **(+0.4)**	93.0 ± 6.8 (-7.0)	83.4 ± 3.0 (-1.5)	37.2 ± 20.8 (+7.3)	4.0	305.9 ± 0.8
Distill (ours)	84.3 ± 4.3 (+9.4)	100.0 ± 0.0 (-0.0)	84.4 ± 0.2 (-0.5)	29.5 ± 5.7 **(-0.4)**	2.6	348.1 ± 5.0

Table 23: Affected-class results on SVHN + ViT-Tiny for partial deletion of digit 9 with class-fraction-to-forget $= 0.5$. Values are mean $\pm$ std over 5 runs, in %. Blue values show deviation from Retrain. Avg. Gap is computed over $\text{RA}_{\text{affected-class}}$, $\text{UA}_{\text{affected-class}}$, and $\text{TA}_{\text{affected-class}}$.

Method	$\text{RA}_{\text{affected-class}}$	$\text{UA}_{\text{affected-class}}$	$\text{TA}_{\text{affected-class}}$	Avg. Gap
Retrain	100.0 ± 0.0	74.9 ± 1.1	72.1 ± 1.4	0
RL	92.5 ± 1.2 (-7.5)	90.3 ± 0.9 (+15.4)	64.9 ± 1.1 (-7.2)	10.0
FT	40.1 ± 48.9 (-59.9)	40.0 ± 48.9 (-34.9)	32.4 ± 39.6 (-39.7)	44.8
GA	100.0 ± 0.0 **(+0.0)**	100.0 ± 0.0 (+25.1)	81.3 ± 0.1 (+9.2)	11.4
IU	100.0 ± 0.0 (+0.0)	100.0 ± 0.0 (+25.1)	81.2 ± 0.0 (+9.1)	11.4
SalUn	88.2 ± 1.2 (-11.8)	86.7 ± 1.1 (+11.8)	61.0 ± 1.3 (-11.1)	11.6
AMUN	83.0 ± 18.5 (-17.0)	75.2 ± 14.0 **(+0.4)**	66.5 ± 12.2 (-5.6)	7.7
Distill (ours)	100.0 ± 0.0 (-0.0)	84.3 ± 4.3 (+9.4)	71.4 ± 3.3 **(-0.7)**	3.4

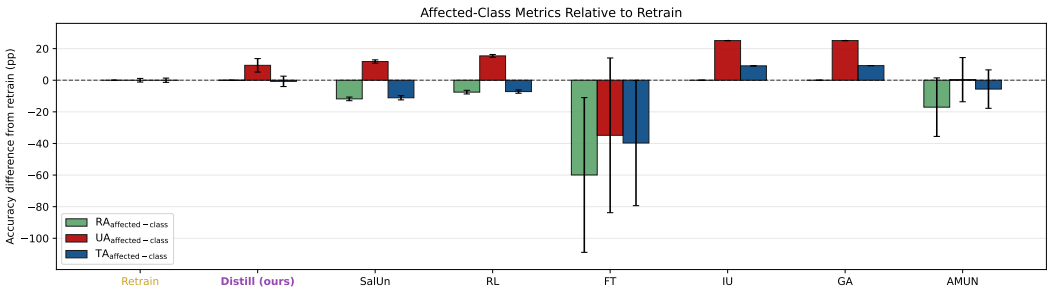

Figure 26: Affected-class deviations from retraining on SVHN + ViT-Tiny for partial deletion of digit 9 with class-fraction-to-forget $= 0.5$. Bars show differences in affected-class RA, UA, and TA relative to Retrain; zero corresponds to exact agreement with the retrained model.

Figure 27 further compares methods across similarity bins. For 50% deletion, LTD stays close to retraining across most retain and test bins, both in accuracy and confidence. By contrast, several baselines have large deviations in the high-similarity region near the forget set. Thus, in this partial-deletion regime, LTD transfers well to SVHN and a transformer backbone.

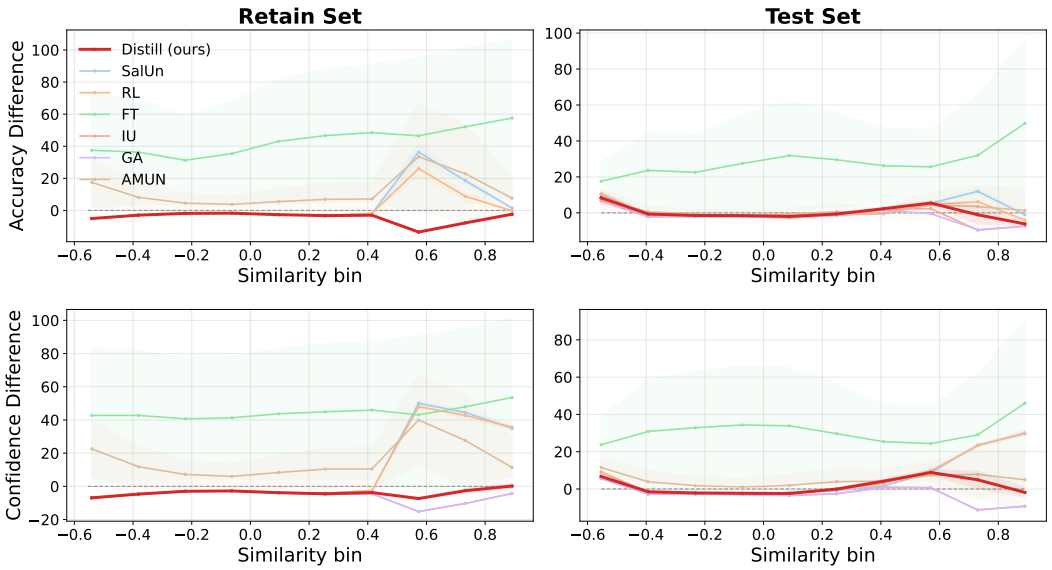

Figure 27: Retrain-relative accuracy and confidence differences across similarity bins on SVHN + ViT-Tiny for class-fraction-to-forget $= 0.5$. Similarity is computed in the penultimate-layer representation space of the fully trained model with respect to the forget set. Top row: accuracy differences; bottom row: correct-class confidence differences. Left: retain set; right: test set. Values close to zero indicate agreement with retraining.

The 90% deletion setting is more challenging (Tables 24 and 25). Here AMUN achieves the lowest global Avg. Gap, while LTD ranks second. LTD closely matches retraining in UA, RA, and TA, but differs more substantially in MIA. In our convention, a higher MIA value means that deleted examples are more often classified as non-members; however, our evaluation is retrain-relative, so the target is to match the auditing behavior of the retrained model rather than to maximize this score. Thus, the larger MIA value of LTD indicates an over-shift relative to retraining in this regime.

The affected-class metrics show a different view. LTD achieves the lowest affected-class Avg. Gap, improving over AMUN (7.3 vs. 12.1), and keeps all three affected-class deviations moderate. AMUN better matches affected-class TA, but has a larger drop in affected-class RA.

Table 24: SVHN + ViT-Tiny results for partial deletion of digit 9 with class-fraction-to-forget $= 0.9$. Values are mean $\pm$ std over 5 runs; accuracy metrics are in $\%$ and RTE is in seconds. Blue values show deviation from Retrain. Avg. Gap is the mean absolute deviation from Retrain over UA, RA, TA, and MIA.

Method	UA	RA	TA	MIA	Avg. Gap	RTE
Retrain	51.1 ± 2.2	100.0 ± 0.0	82.9 ± 1.3	56.2 ± 2.4	0	2058.9 ± 5.9
RL	36.0 ± 4.1 (-15.0)	99.6 ± 0.0 (-0.4)	81.9 ± 0.2 (-1.0)	100.0 ± 0.0 (+43.8)	15.1	200.5 ± 1.8
FT	43.7 ± 46.1 (-7.4)	80.4 ± 20.5 (-19.6)	70.9 ± 15.5 (-12.0)	58.6 ± 47.8 (+2.4)	10.3	182.3 ± 0.6
GA	36.8 ± 28.7 (-14.3)	54.0 ± 22.2 (-46.0)	46.4 ± 19.0 (-36.5)	60.4 ± 24.3 (+4.2)	25.2	7.9 ± 0.1
IU	100.0 ± 0.0 (+48.9)	100.0 ± 0.0 (**+0.0**)	85.2 ± 0.0 (+2.3)	0.0 ± 0.0 (-56.1)	26.8	36.6 ± 0.3
SalUn	31.5 ± 3.8 (-19.6)	99.5 ± 0.0 (-0.5)	81.7 ± 0.2 (-1.1)	100.0 ± 0.0 (+43.8)	16.3	357.6 ± 1.2
AMUN	57.0 ± 15.5 (**+5.9**)	94.1 ± 4.7 (-5.9)	83.4 ± 2.6 (**+0.5**)	57.6 ± 16.7 (**+1.4**)	3.4	393.9 ± 0.8
Distill (ours)	57.6 ± 7.4 (+6.5)	99.9 ± 0.0 (-0.1)	82.2 ± 0.3 (-0.7)	88.1 ± 3.8 (+31.9)	9.8	306.9 ± 2.9

Table 25: Affected-class results on SVHN + ViT-Tiny for partial deletion of digit 9 with class-fraction-to-forget $= 0.9$. Values are mean $\pm$ std over 5 runs, in %. Blue values show deviation from Retrain. Avg. Gap is computed over $\text{RA}_{\text{affected-class}}$, $\text{UA}_{\text{affected-class}}$, and $\text{TA}_{\text{affected-class}}$.

Method	**$\text{RA}_{\text{affected-class}}$**	**$\text{UA}_{\text{affected-class}}$**	**$\text{TA}_{\text{affected-class}}$**	**Avg. Gap**
Retrain	100.0 ± 0.0	51.1 ± 2.2	48.9 ± 2.3	0
RL	36.3 ± 4.3 (-63.7)	36.0 ± 4.1 (-15.0)	23.8 ± 2.6 (-25.1)	34.6
FT	44.5 ± 45.5 (-55.5)	43.7 ± 46.1 (-7.4)	34.5 ± 37.1 (-14.4)	25.8
GA	36.3 ± 28.0 (-63.7)	36.8 ± 28.7 (-14.3)	29.9 ± 24.0 (-19.0)	32.3
IU	100.0 ± 0.0 **(+0.0)**	100.0 ± 0.0 (+48.9)	81.2 ± 0.0 (+32.3)	27.0
SalUn	30.7 ± 3.2 (-69.3)	31.5 ± 3.8 (-19.6)	20.8 ± 2.4 (-28.1)	39.0
AMUN	70.0 ± 21.8 (-30.0)	57.0 ± 15.5 **(+5.9)**	49.4 ± 14.2 **(+0.5)**	12.1
Distill (ours)	93.0 ± 4.1 (-7.0)	57.6 ± 7.4 (+6.5)	40.6 ± 5.1 (-8.3)	7.3

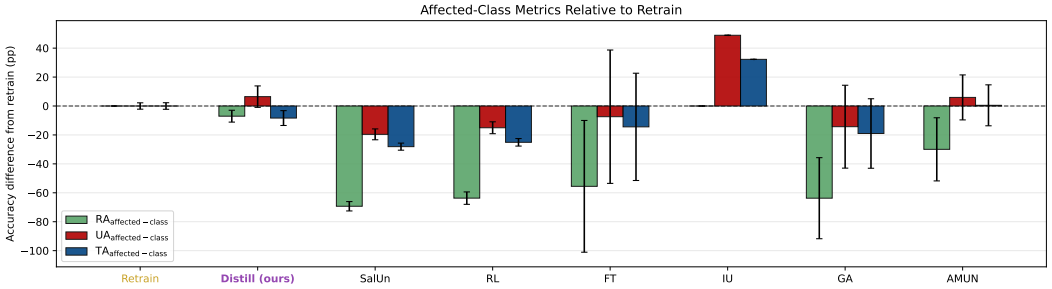

Figure 28: Affected-class deviations from retraining on SVHN + ViT-Tiny for partial deletion of digit 9 with class-fraction-to-forget $= 0.9$. Bars show differences in affected-class RA, UA, and TA relative to Retrain; zero corresponds to exact agreement with the retrained model.

The binned plots in Figure 29 provide a more fine-grained view of this effect. In the aggressive 90% deletion regime, several baselines exhibit large retrain-relative deviations that are concentrated in the high-similarity region near the forget set. LTD substantially reduces this localized deviation: its curves stay close to retraining over low- and medium-similarity bins and increase more mildly near the forget neighborhood. The effect is not eliminated, especially for confidence on the most similar test examples, but it is reduced compared with the stronger localized shifts observed for many baselines.

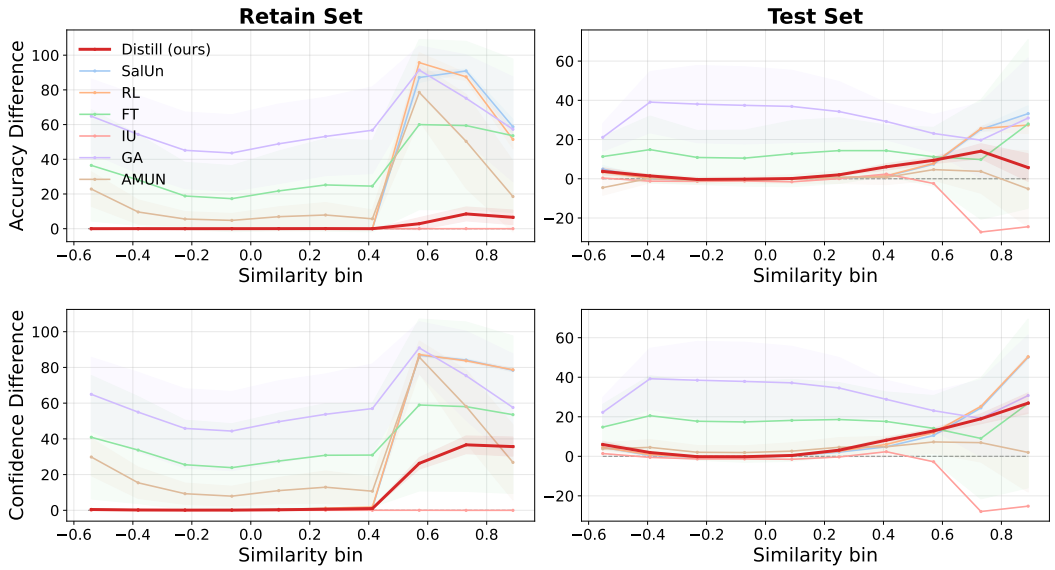

Figure 29: Retrain-relative accuracy and confidence differences across similarity bins on SVHN + ViT-Tiny for class-fraction-to-forget $= 0.9$. Similarity is computed in the penultimate-layer representation space of the fully trained model with respect to the forget set. Top row: accuracy differences; bottom row: correct-class confidence differences. Left: retain set; right: test set. Values close to zero indicate agreement with retraining.

E.4 Heterogeneous forget sets

Our main experiments use partial-class deletion, where D_f is relatively homogeneous and support is selected by similarity to the average forget embedding. When D_f contains several distinct modes, the average embedding may mix different local neighborhoods. In this case, support selection can be made more local. One option is per-example nearest-neighbor selection,

$$S_k(D_f) = \bigcup_{(x_f, y_f) \in D_f} N_k(x_f),$$

where $N_k(x_f) \subset D_r$ contains the k retained examples most similar to x_f in representation space. Alternatively, one can *cluster* D_f and select retained neighbors for each cluster center. These variants preserve the teacher-training and LTD unlearning objectives, but change the support construction and cost.

For clustered support, the teacher can either be trained on the union of the selected neighborhoods or trained separately per cluster. The latter can be parallelized and may be preferable when the clusters correspond to distinct local prediction structures, since a single small teacher may be less effective at fitting all clusters simultaneously.

Support selection can be implemented exactly by scoring all retained examples or approximately using approximate nearest-neighbor search in representation space. Exact average-embedding selection costs $O(|D_r|p)$ after embeddings are computed, while per-example selection costs $O(|D_f||D_r|p)$ without indexing; ANN methods can reduce this cost in large-scale settings.

E.4.1 Experiments on heterogeneous forget set

We additionally evaluate LTD in a heterogeneous setting where the forget set contains examples from multiple classes. Specifically, we remove 50% of class 25 (couch) and 50% of class 0 (apple), and report results averaged over three seeds. For LTD, we use the clustered variant described above in Appendix E.4: forget examples are partitioned into clusters, and a separate local teacher is trained on retained neighbors of each cluster and provides soft targets only to the corresponding forget examples.

Table 26 reports the overall unlearning metrics. Clustered LTD remains close to retraining across UA, RA, and TA, with deviations of $+0.60$, -0.41, and -1.12 percentage points, respectively. This yields an Avg. Gap of 2.35, compared with 10.03 for the closest non-LTD baseline.

Table 26: Overall results for heterogeneous partial-class deletion; the forget set contains 50% of class 25 (couch) and 50% of class 0 (apple). Values are mean ± std over 3 runs, in %. Blue values show deviation from Retrain. Avg. Gap is computed over UA, RA, TA, and MIA.

Method	UA	RA	TA	MIA	Avg. Gap
Retrain	61.67 ± 1.70	99.88 ± 0.02	71.90 ± 0.29	53.87 ± 0.10	0
IU	88.20 ± 4.39 (+26.53)	99.55 ± 0.05 **(-0.33)**	71.50 ± 0.11 **(-0.40)**	38.87 ± 3.77 (-15.00)	10.57
FT	84.27 ± 2.74 (+22.60)	94.50 ± 0.18 (-5.38)	67.51 ± 0.18 (-4.39)	22.47 ± 2.76 (-31.40)	15.94
AMUN	77.00 ± 0.16 (+15.33)	94.18 ± 0.06 (-5.70)	72.83 ± 0.16 (+0.93)	33.20 ± 0.29 (-20.67)	10.66
RL	85.73 ± 3.29 (+24.06)	94.32 ± 0.91 (-5.56)	67.99 ± 0.76 (-3.91)	81.33 ± 8.97 (+27.46)	15.25
SalUn	85.20 ± 3.86 (+23.53)	92.60 ± 1.16 (-7.28)	66.75 ± 0.65 (-5.15)	74.67 ± 11.72 (+20.80)	14.19
GA	46.33 ± 0.34 (-15.34)	93.17 ± 0.02 (-6.71)	65.69 ± 0.04 (-6.21)	65.73 ± 0.38 (+11.86)	10.03
LTD (ours)	62.27 ± 2.39 **(+0.60)**	99.47 ± 0.02 (-0.41)	70.78 ± 0.27 (-1.12)	61.13 ± 5.66 **(+7.26)**	2.35

To verify that this agreement is not obscured by averaging across the two deleted classes, Table 27 reports metrics separately for classes 25 and 0. LTD again gives the smallest aggregate discrepancy from retraining, with an affected-class Avg. Gap of 4.56, compared with 8.41 for the closest baseline. In particular, its retain accuracy remains close to retraining for both classes, while its class-specific forgetting metrics avoid the substantially larger deviations exhibited by most competing methods.

Table 27: Affected-class results for heterogeneous partial-class deletion; the forget set contains 50% of class 25 (couch) and 50% of class 0 (apple). Values are mean ± std over 3 runs, in %. Blue values show deviation from Retrain. Avg. Gap is computed over RA, UA, and TA for both affected classes.

Method	RA_{25}	UA_{25}	TA_{25}	RA_0	UA_0	TA_0	Avg. Gap
Retrain	99.87 ± 0.19	46.80 ± 3.76	51.00 ± 0.82	100.00 ± 0.00	76.53 ± 0.82	78.67 ± 2.62	0
IU	84.53 ± 8.88 (-15.34)	79.47 ± 10.37 (+32.67)	37.33 ± 4.65 (-13.67)	96.53 ± 1.05 (-3.47)	96.93 ± 1.67 (+20.40)	73.33 ± 2.62 (-5.34)	15.15
FT	93.47 ± 4.46 (-6.40)	78.40 ± 5.96 (+31.60)	54.33 ± 7.76 (+3.33)	97.07 ± 0.38 (-2.93)	90.13 ± 0.50 (+13.60)	80.33 ± 3.86 (+1.66)	9.92
AMUN	87.07 ± 0.94 (-12.80)	66.40 ± 1.18 (+19.60)	51.33 ± 1.25 **(+0.33)**	96.67 ± 1.05 (-3.33)	87.60 ± 0.87 (+11.07)	82.00 ± 0.82 (+3.33)	8.41
RL	92.27 ± 4.57 (-7.60)	78.53 ± 7.48 (+31.73)	51.67 ± 10.97 (+0.67)	96.67 ± 2.17 (-3.33)	92.93 ± 1.47 (+16.40)	82.33 ± 5.44 (+3.66)	10.57
SalUn	90.53 ± 4.67 (-9.34)	77.87 ± 5.77 (+31.07)	55.00 ± 5.72 (+4.00)	96.00 ± 2.55 (-4.00)	92.53 ± 4.11 (+16.00)	80.33 ± 7.04 **(+1.66)**	11.01
GA	43.20 ± 0.56 (-56.67)	41.20 ± 0.33 **(-5.60)**	25.00 ± 0.00 (-26.00)	54.00 ± 0.65 (-46.00)	51.47 ± 0.82 (-25.06)	42.00 ± 0.00 (-36.67)	32.67
LTD (ours)	99.33 ± 0.19 **(-0.54)**	53.33 ± 2.78 (+6.53)	45.67 ± 2.87 (-5.33)	97.73 ± 1.85 **(-2.27)**	71.20 ± 3.68 **(-5.33)**	71.33 ± 2.06 (-7.34)	4.56

These results indicate that LTD is not limited to homogeneous partial-class deletion. When distinct modes of the forget set are handled by separate local teachers, the method continues to closely approximate retraining across both aggregate and class-specific metrics.

Ablation of heterogeneous support construction. We next examine which part of the clustered construction is responsible for its improvement. We compare three LTD variants:

- *Pooled*: we average the embeddings of all forget examples, select a single neighborhood using this average vector, and train one teacher on this support;
- *Clustered with a shared teacher*: we select a separate retained neighborhood for each cluster, but train one teacher on the union of the resulting supports;
- *Clustered with separate teachers*: we train a separate local teacher for each cluster, as in the main comparison above.

Tables 28 and 29 report the overall and affected-class results. The pooled construction is substantially weaker, with overall and affected-class Avg. Gaps of 7.51 and 7.64, respectively. In particular, its UA on class 25 differs from retraining by +31.87 percentage points, suggesting that averaging across heterogeneous forget examples can obscure the individual forget-set modes. Nevertheless, the pooled variant remains competitive with the non-LTD baselines, whose smallest corresponding Avg. Gaps are 10.03 and 8.41.

Selecting mode-specific supports reduces the overall Avg. Gap to 3.62 with a shared teacher and 2.35 with separate teachers; the corresponding affected-class Avg. Gaps are 4.67 and 4.56. This suggests that most of the improvement comes from mode-specific support selection, with separate teachers providing a smaller additional benefit.

Table 28: Overall results for the heterogeneous-support ablation. The forget set contains 50% of class 25 (couch) and 50% of class 0 (apple). We compare pooled support selection, cluster-specific supports with a shared teacher, and cluster-specific supports with separate teachers. Values are mean ± std over 3 runs, in %. Blue values show deviation from Retrain. Avg. Gap is computed over UA, RA, TA, and MIA.

Method	UA	RA	TA	MIA	Avg. Gap
Retrain	61.67 ± 1.70	99.88 ± 0.02	71.90 ± 0.29	53.87 ± 0.10	0
LTD (clusters/separate)	62.27 ± 2.39 **(+0.60)**	99.47 ± 0.02 (-0.41)	70.78 ± 0.27 (-1.12)	61.13 ± 5.66 **(+7.26)**	2.35
LTD (clusters/shared)	60.87 ± 1.64 (-0.80)	99.52 ± 0.06 **(-0.36)**	70.86 ± 0.12 **(-1.04)**	66.13 ± 0.19 (+12.26)	3.62
LTD (pooled)	79.73 ± 1.79 (+18.06)	99.40 ± 0.12 (-0.48)	70.53 ± 0.05 (-1.37)	43.73 ± 1.76 (-10.14)	7.51

Table 29: Affected-class results for the heterogeneous-support ablation. The forget set contains 50% of class 25 (couch) and 50% of class 0 (apple). Values are mean ± std over 3 runs, in %. Blue values show deviation from Retrain. Avg. Gap is computed over RA, UA, and TA for both affected classes.

Method	RA_{25}	UA_{25}	TA_{25}	RA_0	UA_0	TA_0	Avg. Gap
Retrain	99.87 ± 0.19	46.80 ± 3.76	51.00 ± 0.82	100.00 ± 0.00	76.53 ± 0.82	78.67 ± 2.62	0
LTD (clusters/separate)	99.33 ± 0.19 **(-0.54)**	53.33 ± 2.78 (+6.53)	45.67 ± 2.87 (-5.33)	97.73 ± 1.85 (-2.27)	71.20 ± 3.68 (-5.33)	71.33 ± 2.06 (-7.34)	4.56
LTD (clusters/shared)	98.53 ± 1.05 (-1.34)	50.67 ± 0.82 **(+3.87)**	41.67 ± 4.03 (-9.33)	98.00 ± 1.31 (-2.00)	71.07 ± 2.96 (-5.46)	72.67 ± 2.50 (-6.00)	4.67
LTD (pooled)	98.93 ± 0.50 (-0.94)	78.67 ± 1.00 (+31.87)	54.33 ± 1.70 **(+3.33)**	98.93 ± 1.00 **(-1.07)**	80.80 ± 2.91 **(+4.27)**	74.33 ± 6.13 **(-4.34)**	7.64

E.5 Class-Conditioned Analysis

To investigate whether the observed similarity-dependent discrepancies are driven solely by class-level effects, we conduct two additional analyses. First, we exclude the affected class from the evaluation. Second, we examine the relationship between similarity and accuracy drop within a single semantically related class (`chair`).

Excluding the affected class. We first repeat the similarity-based analysis on test examples from the remaining 99 classes, excluding the affected class (`couch`). As shown in Table 30, the accuracy drop relative to retraining generally increases with similarity for several baselines, including FT, SalUn, GA, and RL. In contrast, LTD maintains smaller deviations, even in the highest-similarity bin. These results show that localized collateral forgetting is not restricted to examples from the affected class.

Table 30: Similarity-based accuracy differences relative to Retrain, excluding the affected class (`couch`). Results are reported across similarity bins on test examples from the remaining 99 classes. Values are mean ± std over 5 runs, in percentage points, computed as $\mathrm{Acc}_{\mathrm{retrain}} - \mathrm{Acc}_{\mathrm{unlearn}}$. n denotes the number of test examples in each bin.

Bin center	n	IU	FT	AMUN	SalUn	GA	RL	LTD
0.409	187	-1.3 ± 1.6	$+0.4 \pm 2.1$	-1.0 ± 2.1	$+1.3 \pm 1.9$	-1.3 ± 1.9	$+0.6 \pm 2.9$	-0.8 ± 1.3
0.525	3730	$+0.3 \pm 0.3$	$+3.2 \pm 1.0$	-0.5 ± 0.2	$+3.0 \pm 0.8$	$+1.4 \pm 0.2$	$+3.0 \pm 0.7$	$+0.6 \pm 0.4$
0.640	4787	$+0.4 \pm 0.4$	$+5.1 \pm 1.1$	-0.7 ± 0.4	$+4.5 \pm 0.5$	$+2.2 \pm 0.3$	$+4.7 \pm 0.6$	$+1.4 \pm 0.3$
0.756	1089	$+1.1 \pm 0.9$	$+6.5 \pm 1.3$	-1.1 ± 1.2	$+6.0 \pm 1.3$	$+4.9 \pm 0.8$	$+6.2 \pm 1.3$	$+2.0 \pm 0.8$
0.872	107	$+1.3 \pm 3.1$	$+13.8 \pm 4.1$	$+10.3 \pm 4.0$	$+11.0 \pm 6.5$	$+15.1 \pm 2.8$	$+13.1 \pm 4.7$	$+5.2 \pm 6.0$

Within-class analysis. We further restrict the analysis to the semantically related class `chair`, eliminating class identity as an explanation for variation across bins. As shown in Table 31, the accuracy drop generally increases with similarity for several baselines, particularly GA and RL, in the better-populated middle bins ($n = 23, 45, 16$). The extreme bins contain only eight examples each, so we do not draw conclusions from them.

Table 31: Similarity-based accuracy differences relative to Retrain within the `chair` class. Values are mean ± std over 5 runs, in percentage points, computed as $\mathrm{Acc}_{\mathrm{retrain}} - \mathrm{Acc}_{\mathrm{unlearn}}$. n denotes the number of test examples in each bin.

Bin center	n	IU	FT	AMUN	SalUn	GA	RL	LTD
0.688	8	$+2.5 \pm 5.0$	$+2.5 \pm 5.0$	0.0 ± 7.9	$+2.5 \pm 5.0$	$+2.5 \pm 5.0$	0.0 ± 7.9	$+2.5 \pm 5.0$
0.734	23	$+0.9 \pm 3.2$	$+1.7 \pm 4.5$	$+1.7 \pm 2.1$	$+1.7 \pm 2.1$	$+7.0 \pm 2.1$	$+1.7 \pm 2.1$	$+0.9 \pm 3.2$
0.780	45	$+4.9 \pm 1.7$	$+5.3 \pm 4.8$	$+4.0 \pm 3.3$	$+5.3 \pm 3.0$	$+11.6 \pm 2.6$	$+4.9 \pm 3.8$	$+4.0 \pm 4.3$
0.827	16	$+3.8 \pm 6.4$	$+5.0 \pm 7.3$	$+2.5 \pm 7.5$	$+2.5 \pm 6.4$	$+18.8 \pm 3.9$	$+6.2 \pm 3.9$	$+2.5 \pm 6.4$
0.873	8	-20.0 ± 10.0	$+17.5 \pm 10.0$	$+15.0 \pm 9.4$	$+7.5 \pm 10.0$	-7.5 ± 10.0	$+10.0 \pm 12.3$	-12.5 ± 7.9

Together, these analyses provide further evidence that localized collateral forgetting is not simply a class-level effect. The similarity-dependent discrepancy persists after excluding the forget class and is also observed within the semantically related class `chair`, where class identity is fixed. While the extreme bins for `chair` contain too few examples to draw conclusions, the better-populated bins show a pattern consistent with the broader analysis.

F Broader impacts

Machine unlearning can support data-deletion requests and improve transparency in model maintenance by clarifying when unlearning deviates from retraining. At the same time, unreliable approximate unlearning may create a false sense of deletion or privacy protection if used without careful auditing. Our work is diagnostic and methodological, evaluated on public image-classification benchmarks, and should not be interpreted as providing certified deletion guarantees.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the paper's scope and contributions: localized collateral forgetting, theory for GA/RL/RL+FT, and Local Teacher Distillation. These claims are supported by Sections 3–5, while the limitations paragraph in Section 6 states the empirical scope and support-selection assumptions.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper includes a limitations paragraph in Section 6, discussing dependence on the local support set, representation quality, homogeneous forget sets, and the current empirical scope. Computational scaling is discussed in the computational-cost paragraph of Section 4.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Section 3 states the theoretical assumptions and results; Appendix A provides the corresponding proofs.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 5 and Appendix B describe the datasets, baselines, metrics, hyperparameters, and training details; anonymized code is provided.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility.

In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: An anonymized code repository is provided, and the experiments use the publicly available CIFAR-100 dataset. Reproduction details, including training settings and hyperparameter ranges, are given in Section 5 and Appendix B.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 5 describes the dataset, deletion regimes, baselines, and metrics; Appendix B provides training details, optimizers, and hyperparameter ranges.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Main experimental tables and plots report mean $\pm$ standard deviation over 5 runs, and the setup states that results are averaged over 5 runs, with variability computed across independent runs. The MNIST illustration additionally reports 95% confidence intervals.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix B reports the GPU and data-loading setup, and the experimental tables report per-run wall-clock runtimes (RTE).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics; the work uses public datasets and does not involve human subjects or high-risk data/model release.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Appendix F discusses positive impacts for data deletion and model auditing, as well as the risk that approximate unlearning may create a false sense of deletion or privacy protection.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not release high-risk models or scraped datasets; experiments use public image-classification benchmarks and standard model architectures.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The paper cites the datasets, model implementations, and baseline methods used in the experiments; Appendix B lists the main external assets and their licenses where available.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We release anonymized code for the proposed method and experiments, with documentation and instructions in the accompanying repository. The released asset does not introduce new datasets or human-subject data.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing, human-subject experiments, or participant data collection.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing, human-subject experiments, or participant data collection, so IRB approval is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method and experiments do not use LLMs as an important, original, or non-standard component; any LLM use was limited to writing, editing, or formatting assistance.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.