THE SHAPE OF FORGETTING: MODELING RETRAINED OUTPUTS FOR MACHINE UNLEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Machine unlearning is commonly evaluated against an oracle model retrained without the data to be forgotten. In this paper, we empirically investigate whether the outputs of such a retrained model on forget samples can be approximated using only the original model and forget data. This is particularly relevant when retain data, i.e., the original data excluding the forget samples, are no longer available. We study recurring patterns in the transformation from original to retrained outputs by decomposing it into four questions: which forget samples change argmax, which class loses probability mass, how much mass is removed, and where it is redistributed. We observe that uncertainty predicts argmax-change propensity, removed mass concentrates on the original top-1 class in high-accuracy regimes, gained mass favors high-ranked alternatives, and even argmax-stable samples require calibration. Based on these observations, we construct synthetic retrained targets and distill an unlearned model directly on forget data. An extensive set of experiments on Food-101, Oxford Pets, Stanford Cars, CIFAR-100, Tiny-ImageNet with ResNet-18, DeiT-T, and ViT-S evaluate the approach in forget-only regimes with different fractions of removed samples. We show that an unlearned model distilled from the proposed synthetic targets achieves a favorable tradeoff between forgetting, utility, privacy, and distributional alignment: across random 10% and 20% forgetting settings, it tracks the retrained model more closely than competing forget-only baselines in KL/TV on forget samples, remains competitive in test accuracy and membership-inference evaluation, and is substantially faster than retraining.

1 INTRODUCTION

Machine unlearning (MU) aims to remove the contribution of specific training samples without resorting to retraining from scratch (Cadet et al., 2025; Xu et al., 2024; Zhang et al., 2023; Golatkar et al., 2020). MU mechanisms are relevant in several practical settings, most prominently when individuals exercise rights to request the deletion of their personal data under privacy regulations (European Parliament & Council of the European Union, 2016; State of California, 2018). Complete retraining is typically used as the canonical gold-standard reference for unlearning, despite being inefficient and often impractical, or even infeasible when retain data (i.e., training corpus excluding data to forget) are no longer available, as in the setting considered in this paper.

Most classical approaches assume complete or partial access to retain data and rely on it to preserve model utility during unlearning (Wu & Harandi, 2025; Vasilev et al., 2025; Spartalis et al., 2025; Wang et al., 2024; Chundawat et al., 2023a; Kurmanji et al., 2023; Cotogni et al., 2023). In contrast, recent work has investigated retain-data-free settings, where the unlearned model must be obtained without accessing the remaining training data (Cheng et al., 2026; Lee et al., 2026; Kunananthaseelan et al., 2026; Ahmed et al., 2025; Wang et al., 2025; Bonato et al., 2024; Foster et al., 2025). Within this retain-data-free setting, we investigate a different route to approximate retraining: rather than directly designing an update that removes forget-data influence, we ask whether the output distribution that an ideally retrained model would produce on the forget samples can be approximated using only the original model and the forget data. This leads to our central question: *can we approximate retrained-model outputs on forget samples without access to retain data?*

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Typical output-level supervision, such as random relabeling (Golatkar et al., 2020), incompetent teachers (Chundawat et al., 2023a), uniform or smoothed targets (Vasilev et al., 2025; Spartalis et al., 2025), and fixed class suppression (Mora et al., 2026; Dong et al., 2025) do not specify which samples should change class, how much probability should move, or where it should go. For example, penalizing every original top-1 prediction is especially mismatched when only some forget samples change argmax. We instead treat retraining itself as the target object to be approximated. Our empirical analysis decomposes the transition from original to retrain model into a small number of questions about a reasonable synthetic target, and shows that the transition is stochastic but structured: uncertainty can predict likely argmax changes, the original top-1 class is the dominant mass donor, and high-ranked alternatives are the main receivers.

We provide an empirical decomposition under controlled sample-deletion settings and use it to design an informed target transformations. The resulting method, Rank-Spread Flip (RSF), constructs a synthetic retrained teacher on forget samples: stable samples receive mild calibration, while samples selected by an uncertainty gate receive a minimal rank-changing intervention that removes mass from the original top-1 class and redistributes it over high-ranked alternatives. These synthetic outputs are then used as teachers for knowledge-distillation unlearning, allowing the update to mimic a retrained model without accessing the retrained model itself.

Contributions. Our contributions are the following:

- We formulate retain-data-free unlearning target construction as approximating the retrained model, the output of a model retrained without the forget data, using only the original model and forget samples.
- We empirically decompose the original-to-retrained output shift into four target-design questions: which samples change argmax, which class donates probability mass, how much mass should move, and where the mass is redistributed. The analysis shows that retraining is structured but stochastic, motivating uncertainty-based gating, top-1 mass donation, rank-aware redistribution, and stable-sample calibration.
- We instantiate these findings as Rank-Spread Flip, a synthetic target construction used for knowledge distillation on forget data alone. Across CIFAR-100, Tiny-ImageNet, Food-101, Oxford Pets, and Stanford Cars with DeiT-T and ViT-S, RSF improves distributional alignment to retraining while remaining competitive in utility, privacy, and runtime.

We share our code at: https://anonymous.4open.science/r/the-shape-of-forgetting.

2 Related Work

MU methods can be categorized according to the information available during unlearning: methods that access retain data, methods that avoid retain data but rely on proxy data, and methods that operate strictly on forget data (as our approach). We then discuss the methods that perturb supervision on forget samples, most closely related to the method we derive in Section 5. Finally, we point to previous work that demonstrates that outputs of trained model carries membership information.

Ⓐ **Retain-data-aware unlearning.** Many strong unlearning methods explicitly rely on retain data to preserve utility while damaging forget behavior. SCRUB (Kurmanji et al., 2023), SalUn (Fan et al., 2024), DUCK (Cotogni et al., 2023), SSD (Foster et al., 2024), MUNBa (Wu & Harandi, 2025), UNSIR (Tarun et al., 2023), and CoUn (Khalil et al., 2026) instantiate this principle through teacher-student objectives, saliency masks, embedding-space moves, parameter dampening, bargaining objectives, impair-repair updates, or contrastive representation preservation. Sparsity-aware unlearning (Jia et al., 2023) is complementary, showing that pruning can simplify later approximate deletion. These methods uses information unavailable in our retain-data-free formulation.

Ⓑ **Auxiliary-data unlearning.** A second regime avoids the original retain set but introduces generated, out-of-distribution (OOD), or task-specific proxy data. Zero-shot unlearning (Chundawat et al., 2023b), SCAR (Bonato et al., 2024), RUAGO (Lee et al., 2026), ZS-PAG (Chen et al., 2025), and DIET (Kunananthaseelan et al., 2026) use generated noise, OOD distillation, OOD generators, proxy adversarial samples, or small proxy data to protect retain knowledge. These works weaken the retain-data assumption, but still require an external source of information; our method does not.

Forget-data-only unlearning. Closest to our setting are methods that use only the trained model and forget samples. JiT (Foster et al., 2025) suppresses local prediction sensitivity around forget samples; MU-MIs (Cheng et al., 2026) suppresses forget-sample contribution by redirecting sensitivity toward irrelevant classes; TLN (Ye et al., 2025) relabels correctly classified forget samples to the second-ranked class and self-distills; Boundary Unlearning (Chen et al., 2023), instance-wise unlearning (Cha et al., 2024), RCI-Hessian (Mehta et al., 2022), pH-Strips (Qian et al., 2026), and source-free unlearning (Ahmed et al., 2025) act through boundary shifts, adversarial misclassification, localized Hessian updates, feature-subspace removal, or linear-head corrections. These methods share our information constraint, but generally suppress sensitivity, force misclassification, or approximate parameter updates. We instead construct soft forget targets meant to approximate retrained outputs, and distill an unlearned model under this synthetic supervision.

Synthetic supervision. Several unlearning methods define perturbed supervision on forget samples: random labels (Golatkar et al., 2020; Cadet et al., 2025), incompetent teachers (Chundawat et al., 2023a), teacher-student divergence (Kurmanji et al., 2023), logit suppression (Mora et al., 2026), uniform-logit targets (Vasilev et al., 2025), adjusted logits (Dong et al., 2025), reverse-KL distillation (Wang et al., 2024), and Gumbel-based smoothing (Spartalis et al., 2025). These signals are usually paired with retain-data preservation and are often intentionally aggressive. Our distinction is that the forget-side teacher is derived from empirical retraining geometry: only a subset of samples changes argmax, removed mass is mostly top-1 in high-fit regimes, and gained mass follows original rank structure (Section 4).

Output-based memorization signals. Membership-inference and memorization studies show that trained models expose sample-level information through prediction vectors, confidence, entropy, loss, likelihood ratios, or prediction sensitivity (Shokri et al., 2017; Yeom et al., 2018; Song et al., 2019; Song & Mittal, 2021; Carlini et al., 2022; Liu et al., 2022). We use these works only as motivation for an output-space diagnostic: our goal is not to infer membership, but to test whether lightweight, architecture-agnostic outputs are sufficient to approximate retrained forget-sample behavior.

3 Problem Formulation

Reference target. Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ be a supervised classification dataset with labels $y_i \in \mathcal{Y} = \{1, \dots, C\}$. A deletion request defines a forget set $\mathcal{F}$ and retain set $\mathcal{D}_\mathrm{r} = \mathcal{D} \setminus \mathcal{F}$. We denote by O the original model trained on $\mathcal{D}$, and by R the reference model retrained from scratch on $\mathcal{D}_\mathrm{r}$. For sample i, $p_i^O = p^O(x_i)$ and $p_i^R = p^R(x_i)$ are predictive distributions in Δ^{C-1}. We study *output-space unlearning*: construct a target $\hat{p}_i = T(p_i^O; \mathcal{I}_i)$ for each $(x_i, y_i) \in \mathcal{F}$ such that $\hat{p}_i$ approximates p_i^R. The target is not a parameter-level deletion guarantee; it is the teacher distribution later used to train an unlearned model U by distillation (Section 5).

Retraining-induced change. Let $a_i = \arg\max_c p_{i,c}^O$ and $b_i = \arg\max_c p_{i,c}^R$. The binary retraining-change indicator is

$$\chi_i = \mathbb{1}[a_i \neq b_i], \tag{1}$$

and $\hat{\chi}_i = \mathbb{1}[a_i \neq \arg\max_c \hat{p}_{i,c}]$ is the corresponding indicator for the transformed target. Matching retraining therefore requires both distributional closeness and ranking behavior: if $\chi_i = 1$, calibration alone cannot reproduce the retrained top-1 class. The metrics and mass-flow diagnostics used to quantify this approximation are defined in Appendix A.

Retraining reference variability. Standard unlearning evaluation uses a single retrained model R. For diagnostics, however, R is one draw from the randomness of training after deletion. We therefore also train same-split references R_s under different optimization randomness and use

$$\bar{p}_i^R = \frac{1}{S}\sum_{s=1}^{S} p_i^{R_s}, \qquad \pi_i^R = \frac{1}{S}\sum_{s=1}^{S} \mathbb{1}\left[a_i \neq \arg\max_c p_{i,c}^{R_s}\right].$$

Thus $\pi_i^R = 0$ denotes consensus-stable samples, $\pi_i^R = 1$ denotes consensus-changed samples, and $0 < \pi_i^R < 1$ denotes retraining-ambiguous samples.

Table 1: Same-split variability across independent retraining runs. Forget-set accuracy (F Acc.) reports the range across three retraining seeds; all other metrics are averaged over the three pairwise seeds.

Setting	F Acc. range	Argmax agree	Change F_1	TV	sim. KL
CIFAR-100 / ResNet	0.769–0.773	0.810	0.737	0.198	0.345
CIFAR-100 / DeiT-T	0.818–0.827	0.855	0.711	0.147	0.534
T-ImageNet / DeiT-T	0.751–0.752	0.813	0.782	0.193	0.565
T-ImageNet / ViT-S	0.840–0.845	0.871	0.734	0.131	0.429

Table 2: Source of removed probability mass over changed samples. High shares indicate that retraining mostly removes mass from the original top-1 class.

Setting	Acc. O / $\bar{R}$	Change	Top-1 share
CIFAR-100 / ResNet	1.000 / 0.796	0.204	1.000
CIFAR-100 / DeiT-T	1.000 / 0.840	0.160	0.997
T-ImageNet / DeiT-T	0.992 / 0.773	0.224	0.961
T-ImageNet / ViT-S	1.000 / 0.859	0.141	0.998

Information available to the transformation. The paper asks whether T can be built from information available *before* retraining. For a requested record, $\mathcal{I}_i$ may include the original output p_i^O, the forget label y_i, and split-level metadata such as forget-set class counts. It may not include p_i^R, b_i, χ_i, $\bar{p}_i^R$, π_i^R, or any other quantity computed from retrained models. These retrained quantities are used only to analyze what an ideal target should look like; Section 5 then replaces them with deployable rules based on the admissible information.

4 Understanding Retraining by Emprical Observations

To approximate the retrained output p_i^R from p_i^O and $\mathcal{F}$, a target transformation must answer four questions: (1) for which forget samples does retraining change the predicted argmax, (2) which classes lose probability mass, (3) how much mass is removed, and (4) where that mass is redistributed. Below, we provide a set of diagnostics driven by these questions, paired with the prescription they induces for target construction, that we later use to build a synthetic target in Section 5.

Experimental settings. In this section, we use a class-balanced random deletion protocol (details and variants in the appendix), and we focus on classic settings: ResNet-18 trained from scratch, ImageNet-pretrained DeiT-T and ViT-S, on CIFAR-100 and Tiny-ImageNet. If not differently indicated, $\mathcal{F}$ is a random 10% subset of the training data. Diagnostics use the mean retrained output $\bar{p}_i^R$ and change probability π_i^R from Section 3.

Which samples change argmax? Retraining changes the predicted class only for a subset of forget samples, and that subset is not deterministic across retraining seeds. Table 1 shows that same-split retrainings have nearly identical forget accuracy, but their forget-set argmax decisions differ: matching a single retraining seed would partly fit optimization noise. The table reports pairwise retraining agreement on forget samples: argmax agreement and change-mask F_1 measure whether retrained seeds induce the same class transitions, while TV and symmetric KL measure output-distribution variation. We therefore diagnose the stochastic object π_i^R, where stable samples have $\pi_i^R = 0$, changed samples have $\pi_i^R = 1$, and ambiguous samples have $0 < \pi_i^R < 1$. Figure 1 shows that original uncertainty ranks this stochastic transition region. Low-entropy samples are mostly stable, while the highest-entropy quintile contains a large argmax-changing fraction (roughly 40%–65%). The same monotone pattern holds for Tiny-ImageNet / ViT-S (Appendix Figure 4). Appendix C Figure 5 further compares entropy with low confidence, low margin, and their combinations as original-output predictors of retraining-induced argmax changes. **Prescription.** Use original uncertainty as a ranking signal for retraining-instability, and apply an explicit change budget to decide how many forget samples receive an argmax-changing target.

Which class loses mass? If $a_i = \arg\max_c p_{i,c}^O$ is to lose the argmax, probability must leave it. The question is whether retraining removes mass diffusely or mainly from this original top-1 class. Define the class-wise removed mass as $\ell_{i,c} = \max\{0, p_{i,c}^O - \bar{p}_{i,c}^R\}$. For changed samples ($\chi_i = 1$), Table 2 reports the aggregate top-1 donor share

$$\frac{\sum_{i:\chi_i=1} \ell_{i,a_i}}{\sum_{i:\chi_i=1} \sum_c \ell_{i,c}}.$$

The mean-reference changed samples give a simple answer in high-fit regimes: a_i accounts for nearly all removed mass. **Prescription.** Use the original top-1 class as the donor for the argmax-changing branch, consistent with the observed concentration of removed probability mass in the settings studied here.

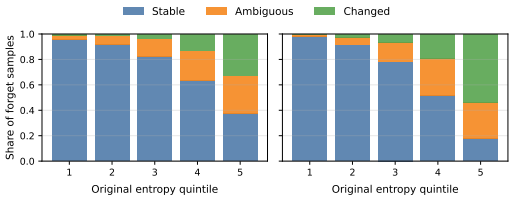


Figure 1: Retraining-change groups across original entropy quintiles. DeiT-T on CIFAR-100 (**left**) and Tiny-ImageNet (**right**). Higher entropy shifts mass from stable to ambiguous and changed samples.

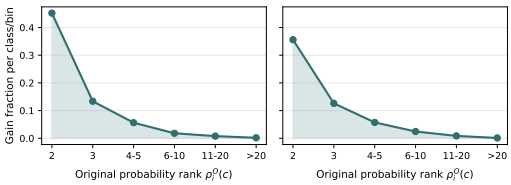


Figure 2: Fraction of positive retraining gain by original probability rank for changed samples. DeiT-T on CIFAR-100 (**left**) and Tiny-ImageNet (**right**). Wide bins are per-class averages. Gain concentrates on high-ranked alternatives.

How much mass should move? After deciding which samples move and which class donates mass, a target must decide how much probability to move. The natural diagnostic analogue is the exact top-1 drop induced by retraining,

$$\bar{d}_i^R = \max\{0,\ p_{i,a_i}^O - \bar{p}_{i,a_i}^R\}.$$

The diagnostics argue against this. Entropy, low confidence, and low margin correlate with $\bar{d}_i^R$ over all forget samples because they identify samples likely to change. But after conditioning on mean-reference changed samples, the signal largely disappears: entropy–mass Spearman correlation is low or reverses (full results in Appendix C Table 7). Uncertainty is a good gate but a poor conditional mass regressor. **Prescription.** Do not fit a per-sample removed mass from confidence; use a simple rank-changing intervention, such as the minimal mass needed to cross the decision boundary.

Where should mass go? The receiving side has a clear structure. Let $\rho_i^O(c)$ be the original-probability rank of class c, where rank 1 is a_i, and let $\bar{v}_{i,c} = \max\{0, \bar{p}_{i,c}^R - p_{i,c}^O\}$ be positive gain. Figure 2 reports the fraction of total gain assigned to original-rank bins on mean-reference changed samples. The runner-up is important but not sufficient: rank 2 receives 0.452 of gained mass on CIFAR-100 and 0.356 on Tiny-ImageNet, while ranks 3–10 also receive substantial mass. Tiny-ImageNet / ViT-S confirms the pattern: $\approx 64\%$ of new top-1 classes lie in original ranks 2–5 and $\approx 76\%$ in ranks 2–10 (full results in Appendix C Table 8). **Prescription.** Use the original non-top-1 tail as a receiver prior: prioritize high-ranked alternatives, while allowing probability mass to spread beyond the runner-up.

Does the output of argmax-stable samples remain unchanged? For samples whose original and retrained argmax agree, retraining can still change the predictive distribution. Table 3 compares identity (no intervention on the output distribution of the original model, $\hat{p}_i = p_i^O$) with full temperature smoothing at different temperatures τ, which softens the original probability vector while preserving the argmax. Identity retains nonzero KL to $\bar{p}_i^R$, whereas mild smoothing captures part of the distributional shift toward retraining; overly large temperatures can over-smooth and worsen the match. **Prescription.** Use mild argmax-preserving calibration for the stable branch, rather than exact identity.

Table 3: Stable-sample calibration on DeiT-Tiny. KL is to $\bar{p}^R$; Moved is $D_{\mathrm{TV}}(p^O, \hat{p})$.

Setting	Target	KL	Moved
CIFAR-100 DeiT-T	Identity	0.381	0.000
	$\tau = 1.5$	0.218	0.023
	$\tau = 2.0$	0.257	0.143
	$\tau = 3.0$	0.841	0.560
T-ImageNet ViT-S	Identity	0.417	0.000
	$\tau = 1.5$	0.228	0.012
	$\tau = 2.0$	0.207	0.085
	$\tau = 3.0$	0.693	0.477

Bridge to target construction. The diagnostics above suggest that approximating retraining is not a single calibration step, but a conditional target-construction problem. First, samples should be ranked by retraining-instability and only a budgeted subset should receive an argmax-changing intervention. Second, when a change in argmax is induced, the original top-1 class is the natural donor, the moved mass should be just large enough to alter the ranking, and the receiver should favor high-ranked non-top-1 alternatives rather than a single class or a uniform tail. Third, samples that remain argmax-stable should still receive mild confidence calibration. Section 5 turns these prescriptions into a practical and deployable transformation.

5 BUILDING FORGET TARGETS FROM ORIGINAL OUTPUTS

Section 4 motivates an output transformation with two branches. The outputs of stable-looking samples benefits from calibration (argmax preserving), while the outputs of unstable-looking samples requires probability mass removal from original top-1 class (to trigger an argmax change) spreading such mass over plausible high-ranked alternatives. We implement this idea with four components: an instability gate g_i, a stable calibrated distribution p_i^S, a moved mass m_i, and a receiver distribution r_i. The synthetic targets $\hat{p}_i$ are then used to distill an unlearned model U on forget data $\mathcal{F}$.

Gate. The first decision is whether a forget sample should be treated as stable or as changing. We use a gate g_i that selects the samples most likely to change argmax under retraining. The gate is entropy-based:

$$g_i(\rho) = \mathbb{1}\left[H(p_i^O) \text{ belongs to the top } \rho|\mathcal{F}| \text{ values over } \mathcal{F}\right]. \tag{2}$$

The parameter $\rho \in [0, 1]$ is the expected forget-change rate. It is the main degree of freedom of the transformation we design: $\rho = 0$ implies no argmax change, while larger ρ moves more forget samples. In practice, ρ can be estimated **from a conservative budget as we show in Section 6 Sensitivity to** ρ. Therefore, the gate selects which transformation to apply as follows:

$$\hat{p}_i = \begin{cases} p_i^S, & g_i = 0, \\ p_i^C, & g_i = 1. \end{cases} \tag{3}$$

Below we explain how stable-sample synthetic targets p_i^S (see stable branch) and changing-sample targets p_i^C (see changing branch) are constructed.

Stable branch. Section 4 shows that the outputs of argmax-stable samples should not be left unchanged: mild temperature smoothing moves their predictive distributions closer to $\bar{p}_i^R$. However, full temperature smoothing spreads probability over all classes, and the stable-sample diagnostic does not tell us how to redistribute that mass class-by-class. We therefore use a conservative variant, *top-1-tail calibration*, which only smooths the binary confidence split between the original top-1 class and the total non-top-1 tail. Appendix C (Table 10) shows the resulting tradeoff: compared with identity it reduces KL to retraining, while compared with full temperature smoothing it keeps TV close to identity and moves much less mass from p_i^O. Let $a_i = \arg\max_c p_{i,c}^O$ be the original top-1 class and let $T_i = 1 - p_{i,a_i}^O$ be the original tail mass. The *top-1-tail calibration* rule first smooths the split between p_{i,a_i}^O and T_i:

$$\tilde{s}_i = \frac{(p_{i,a_i}^O)^{1/\tau_S}}{(p_{i,a_i}^O)^{1/\tau_S} + T_i^{1/\tau_S}}, \qquad \tilde{T}_i = 1 - \tilde{s}_i,$$

and then redistributes $\tilde{T}_i$ proportionally to the original tail:

$$p_{i,a_i}^S = \tilde{s}_i, \qquad p_{i,c}^S = \tilde{T}_i \frac{p_{i,c}^O}{T_i} \quad \text{for } c \neq a_i. \tag{4}$$

When $T_i = 0$, we set $p_i^S = p_i^O$. This branch changes the confidence of the top-1/tail split while preserving the relative geometry inside the original tail and the original argmax.

Changing branch. For samples selected by the gate, the changing branch starts from the stable calibrated distribution p_i^S, removes probability mass from the original top-1 class a_i, and redistributes it to non-top-1 classes. Let e_c denote the one-hot vector on class c, and define the runner-up under the calibrated distribution as

$$c_i^{(2)} = \arg\max_{c \neq a_i} p_{i,c}^S.$$

Given a moved mass m_i and a receiver distribution r_i satisfying $r_{i,a_i} = 0$ and $\sum_c r_{i,c} = 1$, the changing target is

$$p_i^C = p_i^S - m_i e_{a_i} + m_i r_i. \tag{5}$$

The design question is therefore how to choose r_i and m_i. From the empirical analysis in Section 4, we know that the uncertainty of the original outputs is useful to classify which samples should

change, but it is not a reliable conditional predictor of how much top-1 mass retraining removes once a sample is selected. We therefore use a minimal intervention principle: move only enough top-1 mass to make the calibrated runner-up overtake the original top-1 class. To redistribute such mass, we propose the **Rank-Spread Flip (RSF)**, which we explain in the following. RSF uses the calibrated non-top-1 tail itself as the receiver distribution:

$$r_{i,c}^{\mathrm{RSF}} = \frac{p_{i,c}^{S} \mathbb{1}[c \neq a_i]}{1 - p_{i,a_i}^{S}}. \tag{6}$$

This preserves the original rank structure among alternatives while assigning more mass to more plausible classes. The moved mass is the minimal amount that makes the runner-up overtake the donor under this distributed receiver:

$$m_i^{\mathrm{RSF}} = \min \left\{ p_{i,a_i}^{S}, \frac{p_{i,a_i}^{S} - p_{i,c_i^{(2)}}^{S}}{1 + r_{i,c_i^{(2)}}^{\mathrm{RSF}}} + \varepsilon_{\mathrm{num}} \right\}. \tag{7}$$

RSF therefore keeps the intervention minimal while spreading the removed top-1 mass over the high-ranked tail, in line with the empirical observation from Section 4 that retraining mass gains concentrate on high-ranked alternatives but remain distributed beyond the runner-up. Appendix D ablates a runner-up-only flip and oracle variants, showing that merely crossing the nearest boundary is less distributionally faithful than rank-spread redistribution. Appendix Alg. 1 outlines the RSF.

Self-distillation unlearning with RSF synthetic targets. We initialize the unlearned model U from the original model O and train it on the forget set $\mathcal{F}$ via knowledge distillation (Hinton et al., 2015; Gou et al., 2021), using the RSF synthetic targets:

$$\mathcal{L}_U^{\mathrm{F}} = \frac{1}{|\mathcal{F}|} \sum_{i \in \mathcal{F}} D_{\mathrm{KL}}(\hat{p}_i \| p^U(x_i)). \tag{8}$$

6 Experimental Evaluation

Experimental Design

Tasks, Datasets and Models. We study sample unlearning for image classification. Unless stated otherwise, unlearning methods are evaluated in the strict forget-data-only regime: at unlearning time they receive the original model O and the forget data F, but not retain data D_R. We use the random-deletion protocol in (Cadet et al., 2025) with forget sets either 10% or 20% of the training data selected uniformly at random. We use five datasets, which are CIFAR-100 (Krizhevsky, 2009), Tiny-ImageNet (Le & Yang, 2015; Deng et al., 2009), Food-101 (Bossard et al., 2014), Oxford Pets (Parkhi et al., 2012), Stanford Cars (Krause et al., 2013), and two model architectures DeiT-Tiny (Touvron et al., 2021) and ViT-S/16 (Dosovitskiy, 2020).

Baselines. We report results for the original model and the retrained model. We also include **Retrain-KD**, an oracle baseline that distills from the retrained model on forget data, to measure whether distillation on forget data alone can, in principle, be feasible. We compare against **eight** baselines, which include four native or directly applicable retain-data-free methods: **NegGrad**, Random Labels (**RL**) (Golatkar et al., 2020), **JiT** (Foster et al., 2025), and **TLN** (Ye et al., 2025), and four adaptations of methods originally designed with retain-data supervision: **UNDIAL** (Dong et al., 2025), **QUIT** (Mora et al., 2026), **UniLogit** (Vasilev et al., 2025), and **SalUn** (Fan et al., 2024). In these adaptations, we keep the forget mechanism of each method but we remove the usage of retain data (e.g., loss terms that distill from retain data). Hyper-parameter in Appendix F.

Evaluation Metrics. MU methods are assessed along three axes (Fan et al., 2024; Xu et al., 2024): forgetting efficacy, utility preservation and efficiency. We use forget accuracy (**FA**) and **U-MIA** (Cadet et al., 2025) to measure unlearning efficacy, and retain (**RA**) and test accuracy (**TA**) to assess utility preservation. We also report the average gap (**Avg. Gap**) (Fan et al., 2024; Cadet et al., 2025) across FA, U-MIA, RA, and TA. Since our method is designed to minimize the distributional discrepancy with the retrained model, we also measure the **KL** divergence and **TV** distance on the forget data. For the last axis, we report run-time efficiency (**RTE**), defined as the ratio between retraining time and the computational time required for unlearning.

Tiny-ImageNet / DeiT-Tiny

Method	Random Data Forgetting (10%) FA	RA	TA	U-MIA	Avg. Δ	KL	TV	RTE	Random Data Forgetting (20%) FA	RA	TA	U-MIA	Avg. Δ	KL	TV	RTE
Original	99.24 (24.68)	99.15 (0.02)	74.44 (0.16)	64.16 (14.55)	9.87	–	–	–	99.22 (24.94)	99.14 (0.01)	74.44 (0.95)	64.13 (14.29)	10.05	–	–	–
Retrained	$74.56_{\pm 0.08}$ (0.00)	$99.17_{\pm 0.02}$ (0.00)	$74.28_{\pm 0.10}$ (0.00)	$49.61_{\pm 0.08}$ (0.00)	0.00	0.00	0.00	1.00	$74.27_{\pm 0.11}$ (0.00)	$99.15_{\pm 0.03}$ (0.00)	$73.49_{\pm 0.07}$ (0.00)	$49.84_{\pm 0.15}$ (0.00)	0.00	0.00	0.00	1.00
Retrain-KD	$75.05_{\pm 0.34}$ (0.49)	$89.64_{\pm 0.51}$ (9.53)	$71.31_{\pm 0.19}$ (2.97)	$51.89_{\pm 0.19}$ (2.27)	3.83	0.15	0.11	7.21	$74.61_{\pm 0.17}$ (0.33)	$83.62_{\pm 0.50}$ (15.53)	$70.00_{\pm 0.53}$ (3.49)	$52.41_{\pm 0.04}$ (2.57)	5.51	0.21	0.13	6.22
QUIT†	$70.76_{\pm 1.07}$ (3.80)	$75.19_{\pm 1.04}$ (23.98)	$57.05_{\pm 0.58}$ (17.23)	$56.33_{\pm 0.18}$ (6.71)	12.90	1.86	0.76	7.70	$70.57_{\pm 0.33}$ (3.70)	$73.73_{\pm 0.76}$ (25.42)	$56.91_{\pm 0.37}$ (16.58)	$56.13_{\pm 0.06}$ (6.29)	13.00	1.90	0.77	7.92
UniLogit†	$0.33_{\pm 0.23}$ (74.23)	$0.45_{\pm 0.24}$ (98.72)	$0.44_{\pm 0.25}$ (73.85)	$49.84_{\pm 0.46}$ (0.23)	61.73	5.18	0.96	9.65	$0.24_{\pm 0.12}$ (74.03)	$0.33_{\pm 0.15}$ (98.82)	$0.31_{\pm 0.12}$ (73.18)	$49.93_{\pm 0.61}$ (0.09)	61.53	5.32	0.96	7.90
UNDIAL†	$70.33_{\pm 3.15}$ (4.23)	$73.85_{\pm 2.60}$ (25.32)	$57.95_{\pm 2.82}$ (16.34)	$55.55_{\pm 0.13}$ (5.94)	12.93	1.88	0.76	10.74	$76.17_{\pm 1.05}$ (1.90)	$77.32_{\pm 1.06}$ (21.83)	$60.17_{\pm 1.59}$ (13.32)	$56.07_{\pm 0.21}$ (6.23)	10.78	1.73	0.73	7.91
SalUn†	$3.98_{\pm 0.04}$ (70.57)	$4.29_{\pm 0.10}$ (94.88)	$4.03_{\pm 0.21}$ (70.25)	$50.56_{\pm 0.30}$ (0.95)	59.14	4.66	0.95	10.58	$4.40_{\pm 0.95}$ (69.87)	$4.49_{\pm 1.01}$ (94.66)	$4.09_{\pm 0.92}$ (69.40)	$50.95_{\pm 0.32}$ (1.11)	58.76	4.64	0.95	9.30
NegGrad	$3.81_{\pm 3.43}$ (70.75)	$3.90_{\pm 3.31}$ (95.27)	$3.63_{\pm 3.08}$ (70.65)	$50.03_{\pm 0.70}$ (0.42)	59.25	10.08	0.96	12.89	$0.51_{\pm 0.04}$ (73.77)	$0.49_{\pm 0.01}$ (98.66)	$0.50_{\pm 0.01}$ (72.99)	$50.30_{\pm 0.31}$ (0.47)	61.47	21.21	0.99	7.93
RL	$2.74_{\pm 0.40}$ (71.82)	$3.07_{\pm 0.31}$ (96.10)	$3.01_{\pm 0.33}$ (71.28)	$50.95_{\pm 0.06}$ (1.33)	60.11	4.69	0.95	11.69	$2.92_{\pm 0.66}$ (71.36)	$3.05_{\pm 0.75}$ (96.10)	$2.93_{\pm 0.65}$ (70.57)	$50.82_{\pm 0.20}$ (0.99)	59.75	4.68	0.95	10.76
TLN	$73.30_{\pm 6.21}$ (1.26)	$75.13_{\pm 5.70}$ (24.04)	$59.98_{\pm 4.09}$ (14.30)	$54.78_{\pm 0.59}$ (5.16)	12.00	1.47	0.66	10.47	$73.16_{\pm 2.14}$ (1.11)	$73.34_{\pm 2.08}$ (25.81)	$60.56_{\pm 1.05}$ (12.93)	$54.57_{\pm 0.54}$ (4.74)	11.32	1.53	0.69	10.23
JiT	$73.27_{\pm 0.32}$ (1.29)	$73.28_{\pm 0.31}$ (25.89)	$60.23_{\pm 0.27}$ (14.06)	$56.54_{\pm 0.38}$ (6.92)	12.01	1.50	0.36	8.90	$74.13_{\pm 0.89}$ (0.14)	$73.63_{\pm 0.90}$ (25.53)	$60.45_{\pm 0.54}$ (13.04)	$56.90_{\pm 0.13}$ (7.06)	11.54	1.46	0.35	5.86
RSF	$74.69_{\pm 0.51}$ (0.13)	$88.83_{\pm 0.56}$ (10.34)	$69.88_{\pm 0.34}$ (4.40)	$51.71_{\pm 0.28}$ (2.10)	4.28	0.87	0.35	7.21	$72.70_{\pm 0.69}$ (1.57)	$86.40_{\pm 0.32}$ (12.76)	$68.08_{\pm 0.35}$ (5.41)	$51.37_{\pm 0.15}$ (1.53)	5.31	0.96	0.41	4.48

Food-101 / ViT-S

Method	Random Data Forgetting (10%) FA	RA	TA	U-MIA	Avg. Δ	KL	TV	RTE	Random Data Forgetting (20%) FA	RA	TA	U-MIA	Avg. Δ	KL	TV	RTE
Original	99.61 (14.50)	99.68 (0.03)	89.30 (0.24)	60.75 (10.86)	6.40	–	–	–	99.64 (15.18)	99.69 (0.05)	89.30 (0.61)	60.66 (10.85)	6.64	–	–	–
Retrained	$85.11_{\pm 0.07}$ (0.00)	$99.65_{\pm 0.02}$ (0.00)	$89.07_{\pm 0.02}$ (0.00)	$49.89_{\pm 0.44}$ (0.00)	0.00	0.00	0.00	1.00	$84.46_{\pm 0.21}$ (0.00)	$99.64_{\pm 0.01}$ (0.00)	$88.69_{\pm 0.07}$ (0.00)	$49.81_{\pm 0.25}$ (0.00)	0.00	0.00	0.00	1.00
Retrain-KD	$85.35_{\pm 0.14}$ (0.24)	$94.33_{\pm 0.62}$ (5.32)	$85.56_{\pm 0.34}$ (3.51)	$53.77_{\pm 0.24}$ (3.87)	3.24	0.13	0.08	6.46	$84.46_{\pm 0.16}$ (0.01)	$95.03_{\pm 0.35}$ (4.61)	$86.28_{\pm 0.17}$ (2.41)	$52.68_{\pm 0.17}$ (2.87)	2.50	0.09	0.07	2.98
QUIT†	$73.34_{\pm 1.03}$ (11.77)	$77.44_{\pm 0.67}$ (22.21)	$67.97_{\pm 0.73}$ (21.09)	$54.07_{\pm 0.21}$ (4.18)	14.82	1.68	0.75	6.77	$69.63_{\pm 0.96}$ (14.83)	$73.94_{\pm 0.54}$ (25.70)	$62.99_{\pm 0.38}$ (25.70)	$54.34_{\pm 0.25}$ (4.53)	17.72	1.65	0.75	5.07
UniLogit†	$0.34_{\pm 0.06}$ (84.78)	$0.48_{\pm 0.13}$ (99.17)	$0.43_{\pm 0.12}$ (88.64)	$50.24_{\pm 0.16}$ (0.35)	68.24	4.60	0.97	6.13	$0.52_{\pm 0.23}$ (83.95)	$0.59_{\pm 0.16}$ (99.05)	$0.53_{\pm 0.19}$ (88.16)	$49.97_{\pm 0.05}$ (0.16)	67.86	4.37	0.97	4.33
UNDIAL†	$85.19_{\pm 0.74}$ (0.08)	$92.84_{\pm 0.46}$ (6.81)	$80.64_{\pm 0.65}$ (8.43)	$52.93_{\pm 0.36}$ (3.04)	4.71	0.87	0.48	6.14	$86.00_{\pm 0.27}$ (1.54)	$93.00_{\pm 0.16}$ (6.63)	$80.18_{\pm 0.27}$ (8.51)	$53.41_{\pm 0.18}$ (3.60)	5.03	0.87	0.48	4.33
SalUn†	$31.42_{\pm 7.95}$ (53.69)	$31.63_{\pm 8.35}$ (68.03)	$29.97_{\pm 7.50}$ (59.10)	$52.89_{\pm 0.52}$ (3.00)	45.96	3.08	0.80	7.28	$4.99_{\pm 0.70}$ (79.47)	$5.03_{\pm 0.84}$ (94.61)	$5.23_{\pm 0.83}$ (83.46)	$50.42_{\pm 0.23}$ (0.61)	64.57	4.49	0.96	5.70
NegGrad	$83.70_{\pm 17.37}$ (1.41)	$84.02_{\pm 17.96}$ (15.63)	$74.11_{\pm 14.55}$ (14.96)	$58.08_{\pm 0.78}$ (8.18)	12.89	1.59	0.26	5.68	62.80 (21.67)	62.72 (36.92)	56.39 (32.30)	$60.13_{\pm 0.11}$ (10.31)	25.33	3.04	0.43	5.07
RL	$2.31_{\pm 0.80}$ (82.80)	$2.33_{\pm 1.05}$ (97.32)	$2.30_{\pm 0.99}$ (86.77)	$49.96_{\pm 0.53}$ (0.07)	66.74	4.34	0.97	7.32	$1.11_{\pm 0.37}$ (83.35)	$1.18_{\pm 0.33}$ (98.46)	$1.18_{\pm 0.38}$ (87.51)	$50.08_{\pm 0.51}$ (0.27)	67.43	4.36	0.97	5.04
TLN	$84.96_{\pm 0.96}$ (0.15)	$86.26_{\pm 1.41}$ (13.39)	$74.68_{\pm 1.15}$ (14.39)	$56.71_{\pm 0.38}$ (6.81)	8.84	0.92	0.42	6.58	$83.88_{\pm 0.46}$ (0.58)	$87.26_{\pm 0.82}$ (12.38)	$74.79_{\pm 0.49}$ (13.90)	$56.71_{\pm 0.15}$ (6.89)	8.47	0.89	0.43	5.51
JiT	$92.35_{\pm 0.16}$ (7.24)	$91.98_{\pm 0.17}$ (7.67)	$80.22_{\pm 0.12}$ (8.84)	$58.83_{\pm 0.28}$ (8.94)	8.16	0.99	0.20	6.46	$80.18_{\pm 0.60}$ (4.29)	$80.15_{\pm 0.57}$ (19.49)	$71.32_{\pm 0.37}$ (17.37)	$56.75_{\pm 0.14}$ (6.93)	12.05	1.28	0.30	6.18
RSF	$85.70_{\pm 0.47}$ (0.58)	$96.88_{\pm 0.04}$ (2.78)	$85.85_{\pm 0.23}$ (3.21)	$52.36_{\pm 0.24}$ (2.46)	2.25	0.56	0.23	6.45	$83.90_{\pm 0.34}$ (0.57)	$95.33_{\pm 0.07}$ (4.31)	$84.14_{\pm 0.17}$ (4.55)	$50.84_{\pm 0.13}$ (1.02)	2.64	0.62	0.30	4.75

Oxford-IIIT Pets / ViT-S

Method	Random Data Forgetting (10%) FA	RA	TA	U-MIA	Avg. Δ	KL	TV	RTE	Random Data Forgetting (20%) FA	RA	TA	U-MIA	Avg. Δ	KL	TV	RTE
Original	100.00 (6.33)	99.82 (0.10)	91.91 (0.27)	58.79 (9.16)	3.86	–	–	–	99.84 (7.35)	99.84 (0.04)	91.91 (0.87)	59.51 (9.03)	4.23	–	–	–
Retrained	$93.67_{\pm 0.76}$ (0.00)	$99.92_{\pm 0.06}$ (0.00)	$91.64_{\pm 0.34}$ (0.00)	$49.63_{\pm 1.53}$ (0.00)	0.00	0.00	0.00	1.00	$92.49_{\pm 0.49}$ (0.00)	99.88 (0.00)	$91.04_{\pm 0.18}$ (0.00)	$50.48_{\pm 0.59}$ (0.00)	0.00	0.00	0.00	1.00
Retrain-KD	$92.44_{\pm 1.58}$ (1.24)	$97.68_{\pm 0.66}$ (2.24)	$88.15_{\pm 0.22}$ (3.49)	$50.80_{\pm 4.05}$ (1.17)	2.00	0.07	0.06	0.38	$91.05_{\pm 1.36}$ (1.44)	$98.75_{\pm 0.55}$ (1.13)	$89.79_{\pm 0.40}$ (1.25)	$47.98_{\pm 0.43}$ (2.51)	1.52	0.04	0.04	0.61
QUIT†	$51.33_{\pm 7.36}$ (42.34)	$61.73_{\pm 7.55}$ (38.19)	$53.22_{\pm 5.75}$ (38.42)	$48.40_{\pm 3.09}$ (1.22)	30.17	1.60	0.57	3.18	$25.24_{\pm 20.92}$ (67.25)	$31.08_{\pm 25.76}$ (68.80)	$25.26_{\pm 23.55}$ (65.78)	$48.97_{\pm 2.21}$ (1.51)	50.93	2.57	0.88	1.98
UniLogit†	$20.55_{\pm 11.05}$ (73.12)	$25.15_{\pm 12.60}$ (74.77)	$20.21_{\pm 9.96}$ (71.42)	$50.21_{\pm 1.53}$ (0.59)	55.10	3.94	0.79	3.19	$3.35_{\pm 1.38}$ (89.14)	$3.66_{\pm 1.36}$ (96.22)	$3.64_{\pm 1.61}$ (87.40)	$49.52_{\pm 1.50}$ (0.97)	68.52	5.54	0.95	2.70
UNDIAL†	$96.17_{\pm 0.55}$ (2.49)	$96.59_{\pm 1.80}$ (3.33)	$86.44_{\pm 1.45}$ (5.19)	$55.70_{\pm 0.94}$ (6.07)	4.27	0.24	0.18	3.17	$90.73_{\pm 1.15}$ (1.76)	$97.75_{\pm 0.58}$ (2.13)	$89.61_{\pm 1.04}$ (1.43)	$49.61_{\pm 1.15}$ (0.88)	1.64	0.54	0.42	1.98
SalUn†	$93.40_{\pm 5.89}$ (0.28)	$95.37_{\pm 3.08}$ (4.55)	$84.31_{\pm 3.42}$ (7.33)	$55.32_{\pm 1.18}$ (5.70)	5.50	0.31	0.15	3.14	$82.69_{\pm 7.43}$ (9.80)	$84.37_{\pm 8.96}$ (15.51)	$71.66_{\pm 8.76}$ (19.37)	$54.80_{\pm 1.33}$ (4.32)	12.34	0.67	0.28	2.91
NegGrad	$92.65_{\pm 1.69}$ (1.02)	$96.60_{\pm 0.83}$ (3.32)	$86.44_{\pm 0.82}$ (5.19)	$54.31_{\pm 0.42}$ (4.69)	3.79	0.28	0.11	2.75	$77.16_{\pm 15.31}$ (15.34)	$79.74_{\pm 15.10}$ (20.14)	$68.96_{\pm 14.39}$ (22.08)	$52.17_{\pm 1.90}$ (1.69)	14.96	0.92	0.28	2.97
RL	$40.79_{\pm 3.23}$ (52.89)	$45.80_{\pm 3.11}$ (54.12)	$37.09_{\pm 4.37}$ (54.54)	$50.21_{\pm 4.47}$ (0.59)	40.66	2.07	0.73	3.41	$3.94_{\pm 1.04}$ (88.55)	$4.80_{\pm 0.77}$ (95.08)	$3.56_{\pm 0.55}$ (87.48)	$50.54_{\pm 2.61}$ (0.06)	67.89	3.45	0.94	2.97
TLN	$89.46_{\pm 1.15}$ (4.22)	$95.11_{\pm 1.26}$ (4.81)	$83.99_{\pm 1.21}$ (7.65)	$51.49_{\pm 1.22}$ (1.86)	4.76	0.37	0.16	3.08	$91.27_{\pm 2.28}$ (1.22)	$94.62_{\pm 1.12}$ (5.26)	$83.25_{\pm 1.50}$ (7.79)	$54.23_{\pm 1.27}$ (3.74)	4.76	0.39	0.16	2.82
JiT	$93.61_{\pm 0.96}$ (0.06)	$94.73_{\pm 0.05}$ (5.19)	$83.43_{\pm 0.09}$ (8.21)	$56.44_{\pm 0.79}$ (6.82)	5.30	0.32	0.12	3.37	$93.66_{\pm 0.33}$ (1.17)	$94.74_{\pm 0.13}$ (5.14)	$83.76_{\pm 0.30}$ (7.27)	$56.22_{\pm 0.29}$ (5.74)	4.76	0.36	0.13	2.36
RSF	$92.55_{\pm 1.12}$ (1.13)	$97.39_{\pm 0.50}$ (2.53)	$87.45_{\pm 0.98}$ (4.18)	$56.50_{\pm 1.58}$ (6.87)	3.81	0.19	0.10	2.62	$93.18_{\pm 0.51}$ (0.69)	$99.49_{\pm 0.08}$ (0.39)	$90.27_{\pm 0.49}$ (0.77)	$55.34_{\pm 0.36}$ (4.86)	1.61	0.19	0.10	1.67

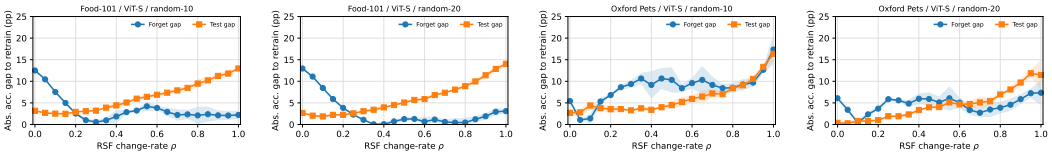
Table 4: Performance summary of MU methods in two unlearning scenarios, 10% random data forgetting and 20% random data forgetting, on Tiny-ImageNet / DeiT-Tiny, Food-101 / ViT-S, and Oxford-IIIT Pets / ViT-S. Entries are reported as $a_{\pm b}$, with mean a and standard deviation b over five independent runs. The absolute performance gap against Retrained is given in blue parentheses. Smaller gaps indicate better alignment with Retrained. Retrain-KD is an oracle baseline that distills directly from the retrained model. Methods marked with † were originally proposed for retain-data-aware settings and are run here without retain data. RTE is the ratio between retraining time and method time, so larger values indicate faster unlearning; the gray row highlights RSF.

Experimental Results

We now evaluate RSF as a forget-only distillation target for training an unlearned model with Eq. 8.

Overall unlearning performance. Table 4 reports the main results for the primary settings, while Appendix Tables 14 report the remaining settings. We compare all methods against Retrain, the gold-standard reference, and include Retrain-KD as an oracle that distills directly from retrained soft targets. RSF provides the most balanced behavior across Avg. Gap and distributional metrics: several baselines can match one isolated quantity, often by over-forgetting or by substantially degrading retain/test utility, but they do not consistently match the retrained model across forgetting, utility, privacy, and output distribution. RSF reaches this tradeoff with competitive runtime efficiency.

Distributional alignment. RSF is motivated by the diagnostics in Section 4, which characterize unlearning as approximating the output distribution of an ideal retrained model. Accordingly, Table 4 and Appendix Table 14 report KL and TV distances to the retrained reference. Retrain-KD provides an oracle reference: when the unlearned model is distilled directly from retrained soft targets, it can closely approximate retrained outputs using only optimization on forget data. Among deployable baselines with competitive Avg. Gap, RSF consistently achieves the strongest distributional alignment, substantially moving the original model toward the retrained output distribution in both KL and TV. This suggests that the RSF target is not only geometrically closer to retraining before optimization, but can also be effectively learned through distillation on forget data alone.



(a) Food-101, 10% (b) Food-101, 20% (c) Oxford Pets, 10% (d) Oxford Pets, 20%

Figure 3: Sensitivity of RSF to the change budget ρ on ViT-S settings. Each panel reports the absolute gap to Retrain in forget accuracy (FA) and test accuracy (TA) as ρ varies, while keeping the KD optimizer, learning rate, epochs, and temperature fixed within the setting. Lower is better. Moderate budgets rapidly reduce the FA gap, whereas the TA gap grows more gradually; large budgets eventually over-forget.

The remaining gap between Retrain-KD and RSF reflects target-construction error. As discussed in Appendix D and Table 11, RSF is inherently imperfect in three places: the entropy gate may select a different change set from retraining, the minimal flip can move less top-1 mass than retraining moves, and the rank-based receiver can differ from the retrained class-wise redistribution. Nevertheless, RSF substantially improves distributional matching on outputs of forget data over generic perturbation targets such as random labels, uniform-logit targets, and unconditioned top-1 penalization. This supports the central premise of our approach: even an approximate model of retraining geometry provides a more faithful forget-side teacher than targets designed only to suppress or confuse the original prediction.

Sensitivity to the change budget ρ. RSF has one fundamental parameter, the target change rate ρ. Figure 3 reports the absolute gaps to retraining in FA and TA for two settings, while Appendix F reports the same and extended diagnostic for all settings. In this analysis, only ρ is varied: the KD optimizer, learning rate, number of epochs, and temperature are fixed within each setting. Thus, the curves should be read as a sensitivity diagnostic rather than as a per-ρ hyperparameter search; better points may exist if the optimizer were retuned separately for each ρ. Increasing ρ from small values to moderate budgets sharply reduces the FA gap, showing that RSF moves quickly from under-forgetting toward retrained forget behavior, while the TA gap grows more gradually. Larger budgets are not expected to help: $\rho > 0.6$ would force rank-changing targets on most forget samples, although retraining changes only a minority of argmax decisions. Consistently, beyond this range RSF enters an over-forgetting regime, where forget alignment stops improving reliably and test utility degrades steadily.

7 Limitations

Our decomposition presented in Section 4 is empirical. RSF assumes that the original top-1 class is a meaningful donor, and, when O is inaccurate on the forget set, a_i and y_i can diverge and top-1 removal may no longer approximate deletion semantics. We also note that rank-based receivers are conservative when retraining sends mass to classes that were not already plausible under O.

8 Conclusion

We studied MU as a target-construction problem: can the retrained output on forget samples be approximated from the original model and the forget data alone? Across controlled image-classification settings, retraining exhibits a structured but stochastic geometry: uncertainty can be a good predictor of samples that will change their predicted argmax, and, for those samples that change, mass is mostly removed from the original top-1 class, and tend to be redistributed according to original rank structure. For those samples that do not change argmax, forms of mild calibration help approximating the distributional matching with the retrained model. These observations shaped our RSF, a mechanism to build a synthetic target for knowledge-distillation unlearning. We showed that RSF-supervised unlearning outperforms state-of-the-art methods when only access to forget data and the original model.

AI USE STATEMENT

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

Additionally, we used generative AI tools for: suggest experimental parameters, create or edit software code, summarize or analyse existing literature, identify relevant literature. We have reviewed all AI-assisted work. Generated code has been manually verified by one author of the paper. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Sk Miraj Ahmed, Umit Yigit Basaran, Dripta S Raychaudhuri, Arindam Dutta, Rohit Kundu, Fahim Faisal Niloy, Basak Guler, and Amit K Roy-Chowdhury. Towards source-free machine unlearning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4948–4957, 2025.

Jacopo Bonato, Marco Cotogni, and Luigi Sabetta. Is retain set all you need in machine unlearning? restoring performance of unlearned models with out-of-distribution images. In *European Conference on Computer Vision*, pp. 1–19. Springer, 2024.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*, pp. 446–461. Springer, 2014. doi: 10.1007/978-3-319-10599-4_29.

Xavier F. Cadet, Anastasia Borovykh, Mohammad Malekzadeh, Sara Ahmadi-Abhari, and Hamed Haddadi. Deep unlearn: Benchmarking machine unlearning for image classification. *arXiv preprint*, 2025. URL https://github.com/xcadet/deepunlearn.

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. In *2022 IEEE symposium on security and privacy (SP)*, pp. 1897–1914. IEEE, 2022.

Sungmin Cha, Sungjun Cho, Dasol Hwang, Honglak Lee, Taesup Moon, and Moontae Lee. Learning to unlearn: Instance-wise unlearning for pre-trained classifiers. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 11186–11194, 2024.

Huiqiang Chen, Tianqing Zhu, Xin Yu, and Wanlei Zhou. Zero-shot machine unlearning with proxy adversarial data generation. In James Kwok (ed.), *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, pp. 339–347. International Joint Conferences on Artificial Intelligence Organization, 8 2025. doi: 10.24963/ijcai.2025/39. URL https://doi.org/10.24963/ijcai.2025/39. Main Track.

Min Chen, Weizhuo Gao, Gaoyang Liu, Kai Peng, and Chen 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)*, pp. 7766–7775. IEEE, 2023.

Xinwen Cheng, Zhehao Huang, Wenxing Zhou, Zhengbao He, Ruikai Yang, Yingwen Wu, and Xiaolin Huang. Remaining-data-free machine unlearning by suppressing sample contribution. In *International Conference on Learning Representations*, volume 2026, pp. 27521–27549, 2026.

Vikram S. Chundawat, Ayush K. Tarun, Murari Mandal, and Mohan Kankanhalli. Can bad teaching induce forgetting? unlearning in deep networks using an incompetent teacher. In *AAAI Conference on Artificial Intelligence*, volume 37, pp. 7210–7217, 2023a. doi: 10.1609/aaai.v37i6.25879. URL https://github.com/vikram2000b/bad-teaching-unlearning.

Vikram S Chundawat, Ayush K Tarun, Murari Mandal, and Mohan Kankanhalli. Zero-shot machine unlearning. *IEEE Transactions on Information Forensics and Security*, 18:2345–2354, 2023b.

Marco Cotogni, Jacopo Bonato, Luigi Sabetta, Francesco Pelosin, and Alessandro Nicolosi. Duck: Distance-based unlearning via centroid kinematics. *arXiv preprint arXiv:2312.02052*, 2023.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255, 2009.

Yijiang River Dong, Hongzhou Lin, Mikhail Belkin, Ramon Huerta, and Ivan Vulić. UNDIAL: Self-distillation with adjusted logits for robust unlearning in large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 8827–8840, 2025. URL https://github.com/dong-river/LLM_unlearning.

Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

European Parliament and Council of the European Union. Regulation (EU) 2016/679 of the European Parliament and of the Council, 2016. URL https://data.europa.eu/eli/reg/2016/679/oj.

Chongyu Fan, Jiancheng Liu, Yihua Zhang, Dennis Wei, Eric Wong, and Sijia Liu. Towards generic and efficient machine unlearning via weight saliency. In *International Conference on Learning Representations*, 2024. URL https://github.com/OPTML-Group/Unlearn-Saliency.

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

Jack Foster, Kyle Fogarty, Stefan Schoepf, Zack Dugue, Cengiz Oztireli, and Alexandra Brintrup. An information theoretic approach to machine unlearning. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=t1utIThKHD.

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9301–9309. IEEE, 2020.

Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International journal of computer vision*, 129(6):1789–1819, 2021.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Jinghan Jia, Jiancheng Liu, Parikshit Ram, Yuguang Yao, Gaowen Liu, Yang Liu, Pranay Sharma, and Sijia Liu. Model sparsity can simplify machine unlearning. *Advances in neural information processing systems*, 36:51584–51605, 2023.

Yasser Khalil, Mehdi Setayesh, and Hongliang Li. Coun: Empowering machine unlearning via contrastive learning. *Advances in Neural Information Processing Systems*, 38:131254–131283, 2026.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, June 2013.

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

Nilakshan Kunananthaseelan, Jing Wu, Trung Le, Gholamreza Haffari, and Mehrtash Harandi. Diet: Machine unlearning on a data-diet. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 22698–22706, 2026.

Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. SCRUB: A tool for selective forgetting of learned behaviors in deep neural networks. In *International Conference on Machine Learning*, 2023. URL https://github.com/meghdadk/SCRUB.

Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

SangYong Lee, Sangjun Chung, and Simon Woo. Ruago: Effective and practical retain-free unlearning via adversarial attack and ood generator. *Advances in Neural Information Processing Systems*, 38:145088–145121, 2026.

Lan Liu, Yi Wang, Gaoyang Liu, Kai Peng, and Chen Wang. Membership inference attacks against machine learning models via prediction sensitivity. *IEEE Transactions on Dependable and Secure Computing*, 20(3):2341–2347, 2022.

Ronak Mehta, Sourav Pal, Vikas Singh, and Sathya N Ravi. Deep unlearning via randomized conditionally independent hessians. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10412–10421. IEEE, 2022.

Alessio Mora, Lorenzo Valerio, Paolo Bellavista, and Andrea Passarella. FedQUIT: On-device federated unlearning via a quasi-competent virtual teacher. *arXiv preprint*, 2026.

Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

Chengyao Qian, Jing Wu, Trung Le, Dinh Phung, and Mehrtash Harandi. ph-strips for selective forgetting: A blunt but fast diagnostic baseline for machine unlearning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3306–3315, June 2026.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pp. 3–18. IEEE, 2017.

Liwei Song and Prateek Mittal. Systematic evaluation of privacy risks of machine learning models. In *30th USENIX Security Symposium (USENIX Security 21)*, pp. 2615–2632, 2021.

Liwei Song, Reza Shokri, and Prateek Mittal. Privacy risks of securing machine learning models against adversarial examples. In Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz (eds.), *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019*, pp. 241–257. ACM, 2019. doi: 10.1145/3319535.3354211. URL https://doi.org/10.1145/3319535.3354211.

Christoforos N. Spartalis, Theodoros Semertzidis, Efstratios Gavves, and Petros Daras. LoTUS: Large-scale machine unlearning with a taste of uncertainty. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. URL https://github.com/cspartalis/LoTUS.

State of California. California consumer privacy act of 2018. https://oag.ca.gov/privacy/ccpa, 2018. Cal. Civ. Code § 1798.105, Right to Delete.

Ayush K Tarun, Vikram S Chundawat, Murari Mandal, and Mohan Kankanhalli. Fast yet effective machine unlearning. *IEEE transactions on neural networks and learning systems*, 35(9):13046–13055, 2023.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pp. 10347–10357. PMLR, 2021.

Stefan Vasilev, Christian Herold, Baohao Liao, Seyyed Hadi Hashemi, Shahram Khadivi, and Christof Monz. Unilogit: Robust machine unlearning for LLMs using uniform-target self-distillation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 22453–22472, 2025. URL https://github.com/eBay/Unilogit.

Bichen Wang, Yuzhe Zi, Yixin Sun, Yanyan Zhao, and Bing Qin. Rkld: Reverse kl-divergence-based knowledge distillation for unlearning personal information in large language models. *arXiv preprint arXiv:2406.01983*, 2024.

Yaxuan Wang, Jiaheng Wei, Yuhao Liu, Jinlong Pang, Quan Liu, Ankit Parag Shah, Yujia Bao, Yang Liu, and Wei Wei. Llm unlearning via loss adjustment with only forget data. In *International Conference on Learning Representations*, volume 2025, pp. 43076–43104, 2025.

Jing Wu and Mehrtash Harandi. Munba: Machine unlearning via nash bargaining. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1–12. IEEE, 2025.

Jie Xu, Zihan Wu, Cong Wang, and Xiaohua Jia. Machine unlearning: Solutions and challenges. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 8(3):2150–2168, 2024. doi: 10.1109/TETCI.2024.3379240.

Shanshan Ye, Jie Lu, and Guangquan Zhang. Towards safe machine unlearning: A paradigm that mitigates performance degradation. In *Proceedings of the ACM on Web Conference 2025*, pp. 4635–4652, 2025.

Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*, pp. 268–282. IEEE, 2018.

Haibo Zhang, Toru Nakamura, Takamasa Isohara, and Kouichi Sakurai. A review on machine unlearning. *SN Computer Science*, 4(4):337, 2023. doi: 10.1007/s42979-023-01767-4.

A Evaluation Metrics and Diagnostic Quantities

For a sample set $\mathcal{S}$, distributional closeness to retraining is measured by

$$\mathcal{L}_{\mathrm{KL}}(\mathcal{S}) = |\mathcal{S}|^{-1} \sum_{i \in \mathcal{S}} D_{\mathrm{KL}}(p_i^R \| \hat{p}_i), \qquad D_{\mathrm{KL}}(p\|q) = \sum_{c=1}^{C} p_c \log(p_c/q_c).$$

We also use total variation $D_{\mathrm{TV}}(p,q) = \frac{1}{2}\sum_{c=1}^{C} |p_c - q_c|$, top-1 agreement with retraining, and precision/recall/F_1 for the predicted change mask $\hat{\chi}_i$. Accuracy gaps are absolute differences from the retrained reference on the same split.

Predictive uncertainty is entropy

$$H(p) = -\sum_{c=1}^{C} p_c \log p_c. \tag{9}$$

For scalar distributions such as entropy values, W_1 is the empirical one-dimensional Wasserstein distance: smaller values indicate more similar empirical distributions along that scalar axis.

For mass-flow diagnostics, $\rho_i^O(c)$ denotes the rank of class c under p_i^O, with rank 1 assigned to the original top-1 class a_i. Positive gain and loss induced by retraining are

$$v_{i,c} = \max\{0,\ p_{i,c}^R - p_{i,c}^O\}, \qquad \ell_{i,c} = \max\{0,\ p_{i,c}^O - p_{i,c}^R\}. \tag{10}$$

Because both distributions lie on the simplex, $\sum_c v_{i,c} = \sum_c \ell_{i,c}$; we call this the redistributed mass. The mass removed from the original top-1 class is

$$d_i^R = \ell_{i,a_i} = \max\{0,\ p_{i,a_i}^O - p_{i,a_i}^R\}. \tag{11}$$

To separate top-class confidence from tail shape, let $c^\star(p) = \arg\max_c p_c$ and define the normalized tail $\bar{p}_c = p_c \mathbb{1}[c \neq c^\star(p)]/(1 - p_{c^\star(p)})$. Tail entropy is $H(\bar{p})$, and the tail-entropy gap compares the synthetic and retrained normalized tails.

B Extended Related Work

MU methods can be categorized according to the information available during unlearning. We organize related work into three regimes: methods that access retain data, methods that avoid retain data but rely on auxiliary or generated data, and methods that operate strictly on forget data. Our approach belongs to the latter category. We conclude by discussing the methods most closely related to ours in terms of mechanism, namely approaches based on perturbing the supervision on forget samples, since Section 5 derives our unlearning procedure as a form of synthetic supervision.

Retain-data-aware unlearning. Classical unlearning methods typically assume access to retain data during the unlearning phase. DUCK (Cotogni et al., 2023) moves embeddings of forget data toward incorrect class centroids while preserving feature geometry of retain data. SalUn (Fan et al., 2024) uses forget data to compute a gradient-based weight-saliency mask, then performs unlearning with randomized labels on forget samples while preserving retain performance with retain data. SSD (Foster et al., 2024) finds parameters disproportionately important for the forget set compared with the overall training data, then dampens those parameters in proportion to their relative forget importance. MUNBa (Wu & Harandi, 2025) formulates unlearning as a two-player Nash bargaining problem between a forgetting objective and a preservation objective. UNSIR (Tarun et al., 2023) performs an impair-repair procedure, using error-maximizing noise to damage forget-class behavior and retain data to repair utility. CoUn (Khalil et al., 2026) uses retain-data contrastive learning and supervised preservation so forget representations move toward semantically similar retain clusters. Sparsity-aware unlearning (Jia et al., 2023) is orthogonal to these data-access regimes: it shows that pruning can simplify subsequent approximate unlearning, e.g., by pruning first and then applying an unlearner. A group of works interleave perturbed supervision on forget data with supervision on retain data (Golatkar et al., 2020; Chundawat et al., 2023a; Wang et al., 2024; Vasilev et al., 2025; Dong et al., 2025; Spartalis et al., 2025; Mora et al., 2026). We discuss their specific supervision-perturbation mechanisms in the final paragraph of this section, as they are the most closely related to our approach from a methodological perspective.

Ⓑ Unlearning via auxiliary data. A second line avoids the original retain set but introduces auxiliary information to protect non-forgotten behavior. Zero-shot unlearning (Chundawat et al., 2023b) removes class information using generated noise/proxy patterns and teacher-student filtering when original training samples are unavailable. SCAR (Bonato et al., 2024) uses metric learning on forget samples and distills the original model on out-of-distribution (OOD) images. RUAGO (Lee et al., 2026) combines adversarial soft forget targets with an OOD generator aligned to the original model. ZS-PAG (Chen et al., 2025) generates proxy adversarial samples to approximate inaccessible retain data and constrain the unlearning update. DIET (Kunananthaseelan et al., 2026) performs retain-free unlearning from a small data diet by pushing forget embeddings toward mismatched hyperbolic prototypes. These approaches weaken the retain-data assumption, but still assume auxiliary data, generated samples, or task-specific proxy structure; our method does not.

Ⓒ Retain-data-free unlearning. Closest to our setting are methods that use only the trained model and the forget samples. JiT (Foster et al., 2025) suppresses local sensitivity around forget samples through input perturbations. MU-MIs (Cheng et al., 2026) suppresses the contribution of each forget point by reducing the model's sensitivity for the sample's original class and redirecting it toward an irrelevant class. TLN (Ye et al., 2025) relabels correctly classified forget samples to the second-ranked class and self-distills with powered cross-entropy. Boundary Unlearning (Chen et al., 2023) shifts the decision boundary of the forget class, either by reassigning forget samples toward neighboring classes or by dispersing their activations through an auxiliary shadow class. Instance-wise unlearning for pretrained classifiers (Cha et al., 2024) intentionally misclassifies forget samples using adversarial examples and weight-importance regularization. RCI-Hessian unlearning (Mehta et al., 2022) selects parameter blocks via perturbation-based conditional-dependence tests and applies a localized Hessian update. pH-Strips (Qian et al., 2026) provides a training-free, retain-set-free diagnostic baseline by estimating the forget feature subspace and removing model sensitivity along it. Ahmed et al. (2025) study source-free unlearning in a setting where the feature extractor is fixed or linearized and only the classifier head is updated; they approximate the parameter correction induced by removing forget samples using Hessian information. These methods share our information constraint more closely, but they generally suppress sensitivity, force misclassification, modify parameter subspaces, or approximate parameter updates. We instead construct soft forget targets intended to approximate retrained outputs.

Synthetic supervision. Teacher-based unlearning methods are typically used within retain-aware unlearning procedures, where perturbed supervision on forget data is interleaved with preservation of knowledge on retain data (e.g., through objectives with separate loss terms for forget and retain samples). Random labels (Golatkar et al., 2020; Cadet et al., 2025), incompetent teachers (Chundawat et al., 2023a), teacher-student divergence maximization (Kurmanji et al., 2023), logit suppression (Mora et al., 2026), uniform-logit targets (Vasilev et al., 2025), adjusted logits (Dong et al., 2025), reverse-KL distillation (Wang et al., 2024), and Gumbel-based output smoothing (Spartalis et al., 2025) all define alternative forget-side supervision signals. **Importantly,** this class of methods does not explicitly aim to approximate the supervision that would be produced by a retrained model on forget data; rather, the forget-side targets are often intentionally more aggressive, over-penalizing the original prediction or confusing the model, knowing that this effect will be counterbalanced by retain-data supervision. Our diagnostics show that this can be mismatched to retraining: only a subset of forget samples changes argmax, and the probability mass removed from the original top prediction is redistributed in a structured, rank-dependent way, as we show in 4.

Output-based memorization and membership signals. Previous work on memorization shows that trained models expose sample-level information through their outputs. Membership-inference attacks exploit confidence, entropy, prediction sensitivity, or other indicators from prediction vectors to distinguish training samples from non-members (Shokri et al., 2017; Yeom et al., 2018; Song et al., 2019; Song & Mittal, 2021; Carlini et al., 2022; Liu et al., 2022).

C Extended Empirical Observations

This section completes the results from Section 4 with empirical observations from additional settings, which are excluded from main text for the sake of space.

Which Samples Change Argmax?. Figure 4 shows retraining-change groups across original entropy quintiles for every balanced-10 setting. Table 5 directly compares entropy, low confidence,

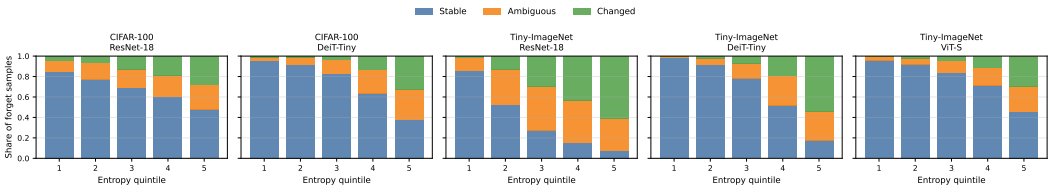


Figure 4: Retraining-change groups across original entropy quintiles for all balanced-10 settings. Stable samples have $\pi_i^R = 0$, changed samples have $\pi_i^R = 1$, and ambiguous samples have $0 < \pi_i^R < 1$. Across architectures, higher entropy shifts mass away from stable samples and toward ambiguous or changed samples.

Table 5: Original-output predictors of mean-reference argmax changes in balanced-10 settings. AUROC is computed against the mask $\mathbb{1}[\arg\max_c p_{i,c}^O \neq \arg\max_c \bar{p}_{i,c}^R]$. Entropy, low confidence, and low margin are consistently informative in high-fit regimes, supporting the entropy gate used by RSF.

Setting	Entropy	Low confidence	Low margin	Change rate
CIFAR-100 / DeiT-Tiny	0.829	0.831	0.831	0.160
Tiny-ImageNet / ResNet-18	0.803	0.811	0.810	0.405
Tiny-ImageNet / DeiT-Tiny	0.871	0.869	0.865	0.224
Tiny-ImageNet / ViT-S	0.818	0.817	0.814	0.141

and low margin as mean-reference argmax-change predictors; they provide comparable AUROC against retrain-induced argmax change. We select entropy as indicator for our method design in Section 5.

Which Class Loses Mass? For each forget sample, define the total mass removed by mean retraining as

$$M_i^- = \sum_{c=1}^{C} \max\{0,\ p_{i,c}^O - \bar{p}_{i,c}^R\}, \tag{12}$$

and the per-sample top-1 source share as

$$s_i^{\text{top1}} = \frac{d_i^R}{M_i^-}, \qquad M_i^- > 0, \tag{13}$$

where d_i^R is defined in Equation 11. The changed-sample restriction is the most relevant one for target construction, because an argmax-changing target must remove enough mass from the original winner a_i to allow another class to become top-1. Table 6 is the expanded version of the donor diagnostic cited in the main text. The top-1 donor assumption is near-exact for high-forget-accuracy settings.

How Much Mass Should Move? Section 4 separates two questions: whether a forget sample should move at all, and how much top-1 mass should be removed after conditioning on movement. The correlations in Table 7 show that original uncertainty (measured as entropy, low confidence, or low margin) is unreliable as a direct conditional mass regressor.

Where Should Mass Go? The main text reports receiver-side rank profiles against the mean retrained reference for two settings (DeiT-T / CIFAR-100 and DeiT-T / Tiny-ImageNet) in Figure 2. Figure 5 reports positive gain as a function of original probability rank across all balanced-10 settings we considered. Table 8 gives the corresponding numeric shares. The pattern is consistent and shows that gained mass concentrates on high-ranked alternatives, but ranks beyond the runner-up receive nontrivial mass. This is the empirical reason for the rank-spread receiver in Section 5.

Argmax-Stable Samples Still Need Calibration.

Table 9 extends the main-text stable-sample diagnostic to additional balanced-10 settings. We restrict to samples whose original and mean-retrained argmax agree, then compare identity targets with full temperature smoothing of p_i^O. Across settings, identity already preserves the argmax but remains

Table 6: Source of probability mass removed by mean retraining in balanced-10 settings. $M_i^- = \sum_c (p_{i,c}^O - \bar{p}_{i,c}^R)_+$ is total removed mass, and s_i^{top1} is the fraction of removed mass coming from the original top-1 class. The changed column conditions on samples whose mean-retrained argmax differs from the original argmax.

Setting	$\lvert\mathcal{F}\rvert$	Acc. $O/\bar{R}$	Change	$\mathbb{E}[M_i^-]$	Agg. top-1 share	$\mathbb{E}[s_i^{\text{top1}}]$	$\mathbb{E}[s_i^{\text{top1}} \mid \chi_i = 1]$
CIFAR-100 / DeiT-Tiny	5000	1.000 / 0.840	0.160	0.186	0.997	0.760	0.997
Tiny-ImageNet / ResNet-18	10000	0.764 / 0.539	0.405	0.401	0.673	0.635	0.633
Tiny-ImageNet / DeiT-Tiny	10000	0.992 / 0.773	0.224	0.253	0.963	0.765	0.961
Tiny-ImageNet / ViT-S	10000	1.000 / 0.859	0.141	0.170	0.997	0.766	0.998

Table 7: Original-output predictors of mean-reference argmax change in balanced-10 settings. Values are Spearman correlations with the mean-retrained top-1 mass removal $\bar{d}_i^R$, reported over all forget samples and only over samples whose mean-retrained argmax differs from the original argmax. The all-sample correlations mainly reflect whether a sample enters the transition region; the changed-only correlations are weak or unstable, so we do not regress a per-sample moved mass from confidence.

Setting	Ent. all	Ent. chg.	Low conf. all	Low conf. chg.	Low margin all	Low margin chg.	Min-flip all	Min-flip chg.
CIFAR-100 / ResNet-18	0.373	0.115	0.382	0.109	0.396	0.098	-0.396	-0.098
CIFAR-100 / DeiT-Tiny	0.515	0.068	0.519	0.062	0.525	0.055	-0.525	-0.055
Tiny-ImageNet / ResNet-18	0.004	-0.883	-0.018	-0.949	-0.019	-0.919	0.019	0.919
Tiny-ImageNet / DeiT-Tiny	0.635	-0.414	0.631	-0.449	0.624	-0.466	-0.624	0.466
Tiny-ImageNet / ViT-S	0.524	0.050	0.523	0.039	0.521	0.025	-0.521	-0.025

separated from $\bar{p}_i^R$ in KL, confirming that retraining changes more than the predicted class. Mild smoothing usually reduces this gap, whereas larger temperatures move substantially farther from the original output and can worsen KL. This supports using a mild, argmax-preserving stable branch rather than leaving stable samples exactly unchanged

Table 10 shows that identity is not an adequate model even on mean-stable samples: KL and TV to $\bar{p}_i^R$ remain nonzero in every setting. Full smoothing often gives the lowest KL, but moves more probability mass. The top-1-tail rule keeps TV close to identity while reducing KL, motivating the tradeoff used by the stable branch in Section 5.

D Rank-Spread vs Runner-Up Flip Ablation

We ablate the changing-branch target construction before training any unlearned model. This target-only diagnostic compares each synthetic target $\hat{p}_i$ directly with the mean retrained output $\bar{p}_i^R$, isolating target geometry from KD optimization. To separate receiver design from gate quality, all flip and oracle rows use the mean-reference change mask

$$\chi_i^{\bar{R}} = \mathbb{1}\left[\arg\max_c p_{i,c}^O \neq \arg\max_c \bar{p}_{i,c}^R\right],$$

which is available only for analysis. Thus, in this diagnostic, the samples treated as changing are fixed, and the question is how well different target geometries match the full retrained output distribution.

The first ablation is *Runner-Up Flip* (RUF), a one-class minimal flip. Using the notation of Section 5, RUF sends all moved mass to the calibrated runner-up $c_i^{(2)}$, i.e.,

$$r_i^{\text{RUF}} = e_{c_i^{(2)}},$$

and moves only enough mass for that class to overtake the original top-1:

$$m_i^{\text{RUF}} = \min\left\{ p_{i,a_i}^S, \frac{p_{i,a_i}^S - p_{i,c_i^{(2)}}^S}{2} + \varepsilon_{\text{num}} \right\}. \tag{14}$$

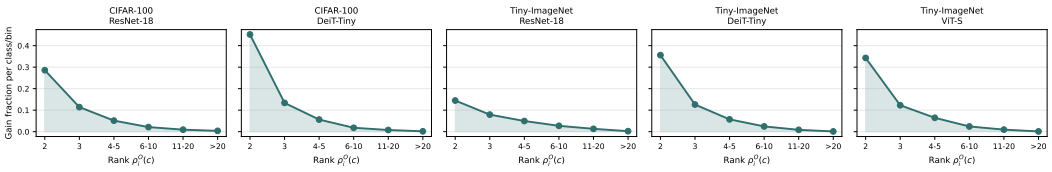


Figure 5: Positive gain under the mean retrained reference as a function of original probability rank for mean-reference changed samples in all balanced-10 settings. The vertical axis is the fraction of total positive gain $(\bar{p}^R_{i,c} - p^O_{i,c})_+$ assigned to each rank bin, divided by the number of classes in that bin. Wide bins 11–20 and > 20 therefore report average gain per class in the bin.

Table 8: Receiver-side rank profile for mean-reference changed samples in balanced-10 settings. Entries are fractions of positive gain assigned to original-probability rank bins; wide bins report aggregate bin mass. The last columns report where the mean-retrained new top-1 class lies in the original ranking.

Setting	Rank 2	Rank 3	4–5	6–10	11–20	> 20	New top-1 2–5	New top-1 2–10
CIFAR-100 / ResNet-18	0.286	0.114	0.103	0.106	0.093	0.298	0.719	0.846
CIFAR-100 / DeiT-Tiny	0.452	0.134	0.113	0.090	0.077	0.134	0.732	0.822
Tiny-ImageNet / ResNet-18	0.145	0.079	0.099	0.136	0.132	0.409	0.630	0.793
Tiny-ImageNet / DeiT-Tiny	0.356	0.126	0.114	0.122	0.085	0.196	0.683	0.812
Tiny-ImageNet / ViT-S	0.343	0.123	0.129	0.121	0.095	0.189	0.641	0.764

RUF tests whether merely crossing the nearest decision boundary is sufficient. RSF uses the same change mask and the same minimal-intervention principle, but spreads moved mass over the calibrated high-ranked tail; it therefore tests whether matching retraining requires distributional redistribution beyond the runner-up.

Table 11 reports the resulting target-only ladder. Calibration alone improves KL but cannot change the argmax. RUF improves top-1 agreement but is too concentrated because all moved mass is assigned to $c_i^{(2)}$. RSF improves distributional matching by spreading mass over plausible high-ranked alternatives, not merely by changing the top-1 class. The oracle rows separate the remaining sources of error. The *Destination* oracle is additionally given the mean-reference winning class, instead of using the calibrated runner-up; the *Destination + Drop* oracle is also given the exact amount of original top-1 mass removed by the mean reference. Even this strongest oracle is not perfect, because it still does not know how retraining distributes probability among the other non-destination classes. At deployment time, RSF replaces $\chi_i^{\bar{R}}$ with the entropy gate $g_i(\rho)$.

Table 11: Target-only diagnostic on all forget samples against the mean retrained reference $\bar{p}_i^R$ for DeiT-Tiny. Stable samples use the top-1-tail calibrated branch; flip rows use the oracle changed set only to isolate receiver quality. Lower KL/TV and higher top-1 agreement with retraining indicate closer approximation.

	CIFAR-100			Tiny-ImageNet		
Target	KL	TV	Top-1 agr.	KL	TV	Top-1 agr.
Calibrated	0.976	0.184	0.840	0.957	0.245	0.776
RUF	0.733	**0.145**	0.917	0.805	0.206	0.871
RSF	**0.479**	0.152	0.917	**0.549**	**0.198**	0.871
Oracles						
Destination	0.409	0.112	1.000	0.469	0.163	0.997
Destination + Drop	0.364	0.080	1.000	0.411	0.127	0.999

E RANK-SPREAD FLIP ALGORITHM

Algorithm 1 summarizes the deployable RSF target construction used in Section 5. The algorithm uses only original probabilities on the forget set and the hyperparameters ρ, τ_S, and ε_{num}. It does not use retrained outputs, retain data, or oracle change masks.

Table 9: Stable-sample full-temperature diagnostic across balanced-10 settings. Stable samples are those whose original and mean-retrained argmax agree. KL/TV are to $\bar{p}^R$; Moved is $D_{\mathrm{TV}}(p^O, \hat{p})$.

Setting	Target	Stable frac.	KL	TV	Moved
CIFAR-100 / DeiT-T	Identity	0.840	0.381	0.057	0.000
	Temp. $\tau = 1.5$	0.840	0.218	0.067	0.023
	Temp. $\tau = 2.0$	0.840	0.257	0.158	0.143
	Temp. $\tau = 3.0$	0.840	0.841	0.547	0.560
Tiny-ImageNet / DeiT-T	Identity	0.776	0.367	0.083	0.000
	Temp. $\tau = 1.5$	0.776	0.207	0.101	0.064
	Temp. $\tau = 2.0$	0.776	0.363	0.262	0.273
	Temp. $\tau = 3.0$	0.776	1.232	0.693	0.718
CIFAR-100 / ResNet-18	Identity	0.796	0.851	0.168	0.000
	Temp. $\tau = 1.5$	0.796	0.352	0.160	0.062
	Temp. $\tau = 2.0$	0.796	0.319	0.242	0.259
	Temp. $\tau = 3.0$	0.796	0.868	0.562	0.664
Tiny-ImageNet / ViT-S	Identity	0.859	0.417	0.055	0.000
	Temp. $\tau = 1.5$	0.859	0.228	0.057	0.012
	Temp. $\tau = 2.0$	0.859	0.207	0.106	0.085
	Temp. $\tau = 3.0$	0.859	0.693	0.465	0.477

Table 10: Argmax-stable calibration diagnostic in balanced-10 settings. Stable samples are those for which the original and mean-retrained argmax agree. Identity keeps p_i^O; full temperature smooths all classes; top-1-tail calibration smooths only the original top-1/tail split. Nonzero identity KL/TV shows that stable samples still undergo distributional calibration under retraining. For full temperature and top-1-tail calibration we use a mild $\tau = 1.5$.

Setting	Target	Stable frac.	F Acc.	KL	TV	Agree	Moved
CIFAR-100 / DeiT-Tiny	identity	0.840	1.000	0.381	0.057	1.000	0.000
	full temperature	0.840	1.000	0.218	0.067	1.000	0.023
	top-1-tail calibration	0.840	1.000	0.254	0.058	1.000	0.007
Tiny-ImageNet / ResNet-18	identity	0.595	0.855	0.661	0.263	1.000	0.000
	full temperature	0.595	0.855	0.410	0.276	1.000	0.222
	top-1-tail calibration	0.595	0.855	0.516	0.252	1.000	0.048
Tiny-ImageNet / DeiT-Tiny	identity	0.776	0.995	0.367	0.083	1.000	0.000
	full temperature	0.776	0.995	0.207	0.101	1.000	0.064
	top-1-tail calibration	0.776	0.995	0.247	0.082	1.000	0.018
Tiny-ImageNet / ViT-S	identity	0.859	1.000	0.417	0.055	1.000	0.000
	full temperature	0.859	1.000	0.228	0.057	1.000	0.012
	top-1-tail calibration	0.859	1.000	0.278	0.055	1.000	0.004

F Extended Experimental Results

Tuning of Original and Retrained Models

Table 12 reports the default configuration for every setting used in Section 6. The data split follows the protocol in (Cadet et al., 2025) and are reported in Table Table 13 . In particular, 15% of training examples are first reserved as a validation set $\mathcal{D}_{\mathrm{v}}$. The remaining 85% are used for training of models. The forget set $\mathcal{F}$ is sampled uniformly as either 10% or 20% among such training data, and the retain set $\mathcal{D}_{\mathrm{r}}$ is the rest of the training corpus. The official test split is held out. The validation set $\mathcal{D}_{\mathrm{v}}$ is used to compute the U-MIA as in (Cadet et al., 2025).

Algorithm 1 Rank-Spread Flip target construction

Require: Original probabilities $\{p_i^O\}_{i\in\mathcal{F}}$, change budget ρ, temperature τ_S, numerical margin ε_{num}

Compute entropy $H_i = -\sum_c p_{i,c}^O \log p_{i,c}^O$
Set $g_i = 1$ for the top $\rho|\mathcal{F}|$ samples by H_i, and $g_i = 0$ otherwise
for each forget sample i **do**
  $a_i \leftarrow \arg\max_c p_{i,c}^O$, $T_i \leftarrow 1 - p_{i,a_i}^O$
  if $T_i = 0$ **then**
    $p_i^S \leftarrow p_i^O$
  else
    $\tilde{s}_i \leftarrow (p_{i,a_i}^O)^{1/\tau_S} / [(p_{i,a_i}^O)^{1/\tau_S} + T_i^{1/\tau_S}]$
    Set $p_{i,a_i}^S \leftarrow \tilde{s}_i$ and $p_{i,c}^S \leftarrow (1-\tilde{s}_i) p_{i,c}^O / T_i$ for $c \neq a_i$
  end if
  if $g_i = 0$ **then**
    $\hat{p}_i \leftarrow p_i^S$
  else
    $r_{i,c} \leftarrow p_{i,c}^S \mathbb{1}[c \neq a_i] / (1 - p_{i,a_i}^S)$
    $c_i^{(2)} \leftarrow \arg\max_{c\neq a_i} p_{i,c}^S$
    $m_i \leftarrow \min\{p_{i,a_i}^S, (p_{i,a_i}^S - p_{i,c_i^{(2)}}^S)/(1 + r_{i,c_i^{(2)}}) + \varepsilon_{\text{num}}\}$
    $\hat{p}_i \leftarrow p_i^S - m_i e_{a_i} + m_i r_i$
  end if
end for
return $\{\hat{p}_i\}_{i\in\mathcal{F}}$

Dataset	Model	Init.	Image	Norm.	Epochs	Batch	Optim.	LR	WD	Warmup
CIFAR-100	ResNet-18	scratch	32×32	CIFAR-100	200	256	SGD	1.0×10^{-1}	5.0×10^{-4}	5
CIFAR-100	DeiT-Tiny	ImageNet	224×224	ImageNet	50	128	AdamW	5.0×10^{-5}	5.0×10^{-2}	5
Tiny-ImageNet	ResNet-18	scratch	64×64	ImageNet	100	256	SGD	1.0×10^{-1}	5.0×10^{-4}	5
Tiny-ImageNet	DeiT-Tiny	ImageNet	224×224	ImageNet	30	128	AdamW	5.0×10^{-5}	5.0×10^{-2}	5
Tiny-ImageNet	ViT-S	ImageNet	224×224	ImageNet	20	128	AdamW	1.0×10^{-4}	5.0×10^{-2}	2
Food-101	ViT-S	ImageNet	224×224	ImageNet	20	128	AdamW	1.0×10^{-4}	5.0×10^{-2}	2
Oxford-IIIT Pets	ViT-S	ImageNet	224×224	ImageNet	20	128	AdamW	1.0×10^{-4}	5.0×10^{-2}	2
Stanford Cars	ViT-S	ImageNet	224×224	ImageNet	20	128	AdamW	1.0×10^{-4}	5.0×10^{-2}	2

Table 12: Default training hyperparameters for reference models. The same configuration is used to train O on the full training pool and R on the retain set after removing $\mathcal{F}$. All runs use cosine learning rate decay.

RSF Change-Budget Sensitivity

RSF uses a single deployment budget ρ, the fraction of forget samples sent to the changing branch. The experiments in this section isolate this parameter: for each setting we fix the RSF target construction and KD optimization hyperparameters used in the corresponding selected row, vary only $\rho \in \{0, 0.05, \ldots, 1\}$, and average over the available same-split training seeds. The shaded regions report one standard deviation. The vertical dashed line marks the retrained argmax-change rate; it is shown only as a diagnostic reference and is not used to choose ρ at deployment.

Figures 6–8 show the accuracy tradeoff induced by ρ, including forget, retain, and test behavior; Figure 9 reports the corresponding KL/TV sensitivity on forget samples. The fixed optimization settings used for the sweeps are the same selected RSF rows reported in the result tables (Tables 4, 14) and in the corresponding hyperparameter tables (Tables 15, 16, 17, 18, 19, and 20).

Dataset	Train source	Test source	Random split	$\lvert\mathcal{D}_{\mathrm{v}}\rvert$	$\lvert\mathcal{F}\rvert$	$\lvert\mathcal{D}_{\mathrm{r}}\rvert$
CIFAR-100	50,000	10,000	10%	7,500	4,250	38,250
CIFAR-100	50,000	10,000	20%	7,500	8,500	34,000
Tiny-ImageNet	100,000	10,000	10%	15,000	8,500	76,500
Tiny-ImageNet	100,000	10,000	20%	15,000	17,000	68,000
Food-101	75,750	25,250	10%	11,362	6,439	57,949
Food-101	75,750	25,250	20%	11,362	12,878	51,510
Oxford-IIIT Pets	3,680	3,669	10%	552	313	2,815
Oxford-IIIT Pets	3,680	3,669	20%	552	626	2,502
Stanford Cars	8,144	8,041	10%	1,222	692	6,230
Stanford Cars	8,144	8,041	20%	1,222	1,384	5,538

Table 13: "Train source" is the official training split before partitioning; "Test source" is the held-out official test split, except for Tiny-ImageNet where the official validation images are used as the final test-style split. We reserve 15% of the training source as validation $\mathcal{D}_{\mathrm{v}}$ before sampling $\mathcal{F}$ from the remaining training pool, in line with the work in (Cadet et al., 2025); the retain set $\mathcal{D}_{\mathrm{r}}$ is the complement. $\mathcal{D}_{\mathrm{v}}$ is used for hyperparameter selection and U-MIA, not for forget-only unlearning updates.

CIFAR-100 / DeiT-Tiny

Method	Random Data Forgetting (10%) FA	RA	TA	U-MIA	Avg. Δ	KL	TV	RTE↑	Random Data Forgetting (20%) FA	RA	TA	U-MIA	Avg. Δ	KL	TV	RTE↑
Original	99.95 (17.55)	99.96 (0.01)	82.04 (0.00)	62.76 (13.05)	7.64	–	–	–	99.95 (18.64)	99.96 (0.01)	82.04 (0.31)	63.33 (13.12)	8.01	–	–	–
Retrained	$82.40_{\pm 0.52}$ (0.00)	$99.98_{\pm 0.01}$ (0.00)	$82.04_{\pm 0.17}$ (0.00)	$49.71_{\pm 0.40}$ (0.00)	0.00	0.00	0.00	1.00	$81.31_{\pm 0.25}$ (0.00)	$99.97_{\pm 0.01}$ (0.00)	$81.73_{\pm 0.20}$ (0.00)	$50.21_{\pm 0.23}$ (0.00)	0.00	0.00	0.00	1.00
Retrain-KD	$82.08_{\pm 0.58}$ (0.32)	$91.36_{\pm 0.81}$ (8.62)	$77.72_{\pm 0.56}$ (4.33)	$51.31_{\pm 3.61}$ (1.60)	3.72	0.08	0.06	10.35	$80.84_{\pm 0.12}$ (0.47)	$90.92_{\pm 0.48}$ (9.05)	$77.73_{\pm 0.49}$ (4.00)	$47.36_{\pm 0.70}$ (2.85)	4.09	0.09	0.06	9.06
QUIT†	$72.65_{\pm 1.00}$ (9.75)	$75.25_{\pm 1.49}$ (24.73)	$61.56_{\pm 1.36}$ (20.48)	$54.44_{\pm 0.32}$ (4.73)	14.94	2.40	0.87	11.10	$73.05_{\pm 0.66}$ (8.26)	$75.80_{\pm 0.61}$ (24.17)	$61.58_{\pm 0.62}$ (20.15)	$54.28_{\pm 0.08}$ (4.07)	14.17	2.25	0.86	9.74
UniLogit†	$1.22_{\pm 0.28}$ (81.18)	$1.23_{\pm 0.28}$ (98.75)	$1.18_{\pm 0.28}$ (80.86)	$50.07_{\pm 0.57}$ (0.36)	65.30	7.80	0.98	14.77	$0.41_{\pm 0.12}$ (80.90)	$0.41_{\pm 0.03}$ (99.56)	$0.33_{\pm 0.05}$ (81.40)	$50.24_{\pm 0.43}$ (0.03)	65.48	4.43	0.97	11.22
UNDIAL†	$82.37_{\pm 0.50}$ (0.03)	$94.62_{\pm 0.48}$ (5.36)	$76.53_{\pm 0.11}$ (5.52)	$53.25_{\pm 0.30}$ (3.54)	3.69	1.12	0.59	11.06	$82.34_{\pm 0.10}$ (1.03)	$94.92_{\pm 0.24}$ (5.05)	$76.57_{\pm 0.49}$ (5.16)	$52.75_{\pm 0.32}$ (2.54)	3.37	1.10	0.59	7.26
SalUn†	$14.13_{\pm 3.70}$ (68.27)	$14.25_{\pm 3.61}$ (85.73)	$12.97_{\pm 3.31}$ (69.07)	$51.08_{\pm 0.77}$ (1.37)	56.13	4.00	0.94	16.04	$18.45_{\pm 5.62}$ (62.86)	$18.36_{\pm 5.61}$ (81.61)	$16.56_{\pm 4.94}$ (65.17)	$51.08_{\pm 0.37}$ (0.87)	52.64	3.77	0.90	12.91
NegGrad	$0.97_{\pm 0.18}$ (81.43)	$1.01_{\pm 0.03}$ (98.97)	1.00 (81.04)	$49.87_{\pm 0.18}$ (0.16)	65.42	27.16	0.99	14.89	$2.50_{\pm 1.59}$ (78.81)	$2.61_{\pm 1.56}$ (97.36)	$2.59_{\pm 1.53}$ (79.14)	$60.22_{\pm 0.39}$ (10.01)	66.34	11.33	0.98	12.82
RL	$3.55_{\pm 0.70}$ (78.85)	$3.46_{\pm 0.57}$ (96.52)	$3.38_{\pm 0.59}$ (78.67)	$50.66_{\pm 1.10}$ (0.95)	63.76	4.41	0.97	17.26	$2.00_{\pm 0.19}$ (79.31)	$1.99_{\pm 0.18}$ (97.98)	$1.94_{\pm 0.43}$ (79.79)	$50.24_{\pm 0.45}$ (0.03)	64.29	4.35	0.97	12.62
TLN	$80.80_{\pm 1.23}$ (1.60)	$82.27_{\pm 1.59}$ (17.71)	$67.61_{\pm 0.98}$ (14.43)	$56.18_{\pm 0.14}$ (6.47)	10.07	1.16	0.30	16.80	$77.36_{\pm 0.99}$ (3.95)	$87.51_{\pm 1.10}$ (12.46)	$69.09_{\pm 0.59}$ (12.64)	$54.27_{\pm 0.11}$ (4.06)	8.29	1.37	0.36	10.91
JiT	$82.59_{\pm 0.84}$ (0.19)	$83.06_{\pm 0.80}$ (16.91)	$68.35_{\pm 0.44}$ (13.69)	$57.46_{\pm 0.11}$ (7.75)	9.71	1.27	0.28	12.79	$80.75_{\pm 0.08}$ (0.56)	$81.13_{\pm 0.08}$ (18.84)	$67.30_{\pm 0.03}$ (14.43)	$57.10_{\pm 0.06}$ (6.88)	10.19	1.34	0.29	8.65
RSF	$80.16_{\pm 0.29}$ (2.24)	$95.62_{\pm 0.45}$ (4.36)	$78.54_{\pm 0.20}$ (3.50)	$52.90_{\pm 0.33}$ (3.18)	3.24	0.83	0.28	10.36	$80.91_{\pm 0.48}$ (0.40)	$93.93_{\pm 0.36}$ (6.04)	$76.63_{\pm 0.37}$ (5.10)	$49.52_{\pm 0.30}$ (0.69)	3.07	0.94	0.37	6.71

Stanford Cars / ViT-S

Method	Random Data Forgetting (10%) FA	RA	TA	U-MIA	Avg. Δ	KL	TV	RTE↑	Random Data Forgetting (20%) FA	RA	TA	U-MIA	Avg. Δ	KL	TV	RTE↑
Original	96.39 (27.37)	96.52 (0.22)	73.50 (2.46)	69.36 (19.77)	12.35	–	–	–	97.04 (31.36)	96.37 (0.20)	73.50 (6.26)	69.57 (19.75)	14.37	–	–	–
Retrained	$69.02_{\pm 0.57}$ (0.00)	$96.74_{\pm 0.14}$ (0.00)	$71.03_{\pm 0.42}$ (0.00)	$49.59_{\pm 1.41}$ (0.00)	0.00	0.00	0.00	1.00	$65.68_{\pm 0.28}$ (0.00)	$96.57_{\pm 0.01}$ (0.00)	$67.23_{\pm 0.10}$ (0.00)	$49.82_{\pm 1.63}$ (0.00)	0.00	0.00	0.00	1.00
Retrain-KD	$63.10_{\pm 1.59}$ (5.92)	$82.67_{\pm 1.54}$ (14.07)	$59.48_{\pm 1.41}$ (11.55)	$55.03_{\pm 1.26}$ (5.44)	9.19	0.33	0.28	2.78	$60.38_{\pm 1.10}$ (5.30)	$79.98_{\pm 0.66}$ (16.58)	$57.56_{\pm 0.22}$ (9.67)	$55.34_{\pm 0.28}$ (5.51)	9.25	0.34	0.29	2.66
QUIT†	$66.18_{\pm 1.30}$ (2.83)	$84.99_{\pm 0.71}$ (11.75)	$59.30_{\pm 0.64}$ (11.74)	$55.56_{\pm 1.50}$ (5.97)	8.16	1.06	0.59	2.46	$77.41_{\pm 1.26}$ (11.73)	$83.76_{\pm 1.10}$ (12.80)	$59.17_{\pm 1.23}$ (8.06)	$61.22_{\pm 0.23}$ (11.40)	10.95	0.95	0.55	2.33
UniLogit†	$0.53_{\pm 0.30}$ (68.49)	$0.69_{\pm 0.22}$ (96.05)	$0.66_{\pm 0.20}$ (70.38)	$49.45_{\pm 2.18}$ (0.14)	58.85	5.01	0.90	3.00	$0.12_{\pm 0.04}$ (65.56)	$0.54_{\pm 0.08}$ (96.02)	$0.50_{\pm 0.12}$ (66.73)	$50.33_{\pm 2.23}$ (0.51)	57.22	3.51	0.82	2.32
UNDIAL†	$66.04_{\pm 0.95}$ (2.98)	$85.01_{\pm 0.57}$ (11.73)	$59.34_{\pm 0.77}$ (11.69)	$56.14_{\pm 0.88}$ (6.55)	8.33	1.05	0.59	2.45	$64.84_{\pm 1.08}$ (0.84)	$82.03_{\pm 0.80}$ (14.53)	$58.17_{\pm 0.38}$ (9.06)	$57.00_{\pm 0.77}$ (7.18)	7.96	0.94	0.55	2.32
SalUn†	$44.46_{\pm 8.66}$ (24.56)	$46.72_{\pm 8.31}$ (50.02)	$29.90_{\pm 5.28}$ (41.13)	$59.20_{\pm 1.71}$ (9.61)	31.42	1.63	0.62	2.88	$52.72_{\pm 11.52}$ (12.96)	$54.88_{\pm 11.49}$ (41.69)	$34.42_{\pm 7.93}$ (32.81)	$60.80_{\pm 0.82}$ (10.97)	24.63	1.41	0.57	2.72
NegGrad	$19.80_{\pm 6.97}$ (49.22)	$22.94_{\pm 7.33}$ (73.80)	$14.84_{\pm 4.38}$ (56.20)	$54.79_{\pm 0.47}$ (5.20)	46.19	2.74	0.75	3.20	$53.88_{\pm 14.78}$ (11.80)	$57.61_{\pm 14.37}$ (38.96)	$37.07_{\pm 10.09}$ (30.16)	$60.62_{\pm 2.02}$ (10.80)	23.71	1.39	0.54	2.15
RL	$0.58_{\pm 0.29}$ (68.44)	$0.47_{\pm 0.10}$ (96.27)	$0.46_{\pm 0.07}$ (70.58)	$49.06_{\pm 1.99}$ (0.53)	59.04	3.42	0.83	2.44	$0.92_{\pm 0.18}$ (64.76)	$0.99_{\pm 0.17}$ (95.57)	$0.87_{\pm 0.20}$ (66.36)	$49.52_{\pm 2.31}$ (0.30)	56.77	3.17	0.80	2.47
TLN	$70.13_{\pm 1.55}$ (1.12)	$86.01_{\pm 0.86}$ (10.72)	$59.56_{\pm 0.79}$ (11.48)	$58.86_{\pm 0.22}$ (9.27)	8.16	0.98	0.43	2.92	$65.37_{\pm 1.15}$ (0.31)	$79.04_{\pm 2.00}$ (17.53)	$54.01_{\pm 1.76}$ (13.22)	$57.82_{\pm 1.42}$ (8.00)	9.93	1.19	0.48	2.63
JiT	$65.85_{\pm 1.37}$ (3.17)	$68.67_{\pm 0.96}$ (28.07)	$43.63_{\pm 0.70}$ (27.40)	$61.97_{\pm 0.54}$ (12.38)	17.84	1.02	0.46	2.96	$60.21_{\pm 0.59}$ (5.47)	$60.65_{\pm 0.34}$ (35.91)	$38.71_{\pm 0.16}$ (28.52)	$62.13_{\pm 0.20}$ (12.31)	20.57	1.19	0.51	2.65
RSF	$68.79_{\pm 1.01}$ (0.23)	$85.97_{\pm 0.59}$ (10.77)	$62.36_{\pm 0.77}$ (8.68)	$58.29_{\pm 0.29}$ (8.69)	7.20	0.58	0.35	2.10	$67.20_{\pm 0.22}$ (1.52)	$83.96_{\pm 0.21}$ (12.61)	$60.10_{\pm 0.43}$ (7.13)	$57.38_{\pm 0.35}$ (7.56)	7.15	0.59	0.36	1.67

Table 14: Performance summary of MU methods in two unlearning scenarios, 10% random data forgetting and 20% random data forgetting, on CIFAR-100 / DeiT-Tiny and Stanford Cars / ViT-S. Entries are reported as $a_{\pm b}$, with mean a and standard deviation b over independent runs. The absolute performance gap against Retrained is given in blue parentheses. Smaller gaps indicate better alignment with Retrained. Retrain-KD is an oracle baseline that distills directly from the retrained model. Methods marked with † were originally proposed for retain-data-aware settings and are run here without retain data. RTE is the ratio between retraining time and method time, so larger values indicate faster unlearning; the gray row highlights RSF.

(a) random-10

Method	Opt.	Ep.	LR	WD	Method-specific HPs
Retrain-KD	adamw	10	1e-04	0	–
QUIT†	adamw	10	1e-05	0	tau=1, variant=logits_min
UniLogit†	adamw	3	1e-05	0	tau=1
UNDIAL†	adamw	10	1e-04	0	γ=8, tau=1
SalUn†	adamw	1	3e-05	0	mask=0.3
NegGrad	adamw	5	3e-04	0	–
RL	adamw	1	3e-05	0	–
TLN	adam	1	1e-05	1e-05	delta=0, lambda=1
JiT	sgd	1	0.022	0	sigma=0.09, pert.=25
RSF	adamw	10	1e-04	0	ρ=0.3

(b) random-20

Method	Opt.	Ep.	LR	WD	Method-specific HPs
Retrain-KD	adamw	10	1e-04	0	–
QUIT†	adamw	5	3e-05	0	tau=1, variant=logits_min
UniLogit†	adamw	3	1e-04	0	tau=1
UNDIAL†	adamw	10	1e-04	0	γ=8, tau=1
SalUn†	adamw	1	3e-05	0	mask=0.05
NegGrad	adamw	3	3e-06	0	–
RL	adamw	3	3e-05	0	–
TLN	adamw	5	3e-05	0	delta=0.5, lambda=0.5
JiT	sgd	5	1e-04	0	sigma=0.05, pert.=3
RSF	adamw	10	1e-04	0	ρ=0.45

Table 15: Selected hyperparameters for CIFAR-100 / DeiT-Tiny.

(a) random-10

Method	Opt.	Ep.	LR	WD	Method-specific HPs
Retrain-KD	adamw	10	3e-05	0	–
QUIT†	adamw	10	1e-05	0	tau=1, variant=logits_min
UniLogit†	adamw	5	3e-05	0	tau=1
UNDIAL†	adamw	3	3e-05	0	γ=12, tau=1
SalUn†	adamw	3	1e-05	0	mask=0.3
NegGrad	adamw	1	1e-05	0	–
RL	adamw	3	1e-05	0	–
TLN	adam	5	3e-05	1e-05	delta=0.005, lambda=3
JiT	sgd	1	0.006	0	sigma=0.1, pert.=25
RSF	adamw	10	1e-04	0	ρ=0.5

(b) random-20

Method	Opt.	Ep.	LR	WD	Method-specific HPs
Retrain-KD	adamw	5	1e-04	0	–
QUIT†	adamw	3	3e-05	0	tau=1, variant=logits_min
UniLogit†	adamw	3	1e-05	0	tau=1
UNDIAL†	adamw	3	1e-05	0	γ=10, tau=1
SalUn†	adamw	1	1e-05	0	mask=0.3
NegGrad	adamw	5	1e-05	0	–
RL	adamw	1	1e-05	0	–
TLN	adam	1	1e-04	1e-05	delta=0.05, lambda=3
JiT	sgd	1	0.003	0	sigma=0.1, pert.=25
RSF	adamw	10	1e-04	0	ρ=0.6

Table 16: Selected hyperparameters for Tiny-ImageNet / DeiT-Tiny.

(a) random-10

Method	Opt.	Ep.	LR	WD	Method-specific HPs
Retrain-KD	adamw	3	1e-04	0.05	–
QUIT†	adamw	3	3e-05	0	tau=1, variant=logits_min
UniLogit†	adamw	5	3e-05	0	tau=1
UNDIAL†	adamw	5	3e-05	0	γ=10, tau=1
SalUn†	adamw	1	3e-05	0	mask=0.05
NegGrad	adamw	10	1e-06	0	–
RL	adamw	3	3e-05	0	–
TLN	adamw	5	1e-05	0	delta=0.5, lambda=1
JiT	adamw	3	1e-06	0	sigma=0.05, pert.=3
RSF	adamw	3	1e-04	0	ρ=0.3

(b) random-20

Method	Opt.	Ep.	LR	WD	Method-specific HPs
Retrain-KD	adamw	10	3e-05	0.05	–
QUIT†	adamw	3	3e-05	0	tau=1, variant=logits_min
UniLogit†	adamw	5	1e-04	0	tau=1
UNDIAL†	adamw	5	3e-05	0	γ=10, tau=1
SalUn†	adamw	1	3e-05	0	mask=0.05
NegGrad	adamw	5	1e-06	0	–
RL	adamw	5	3e-05	0	–
TLN	adamw	3	3e-05	0	delta=0.5, lambda=1
JiT	sgd	1	0.001	0	sigma=0.05, pert.=3
RSF	adamw	3	1e-04	0	ρ=0.45

Table 17: Selected hyperparameters for Food-101 / ViT-S.

(a) random-10

Method	Opt.	Ep.	LR	WD	Method-specific HPs
Retrain-KD	adamw	5	1e-04	0	–
QUIT†	adamw	3	3e-05	0	tau=1, variant=logits_min
UniLogit†	adamw	3	3e-05	0	tau=1
UNDIAL†	adamw	3	1e-04	0	γ=4, tau=1
SalUn†	adamw	3	3e-05	0	mask=0.2
NegGrad	adamw	10	3e-06	0	–
RL	adamw	3	3e-05	0	–
TLN	adamw	5	1e-05	0	delta=0.5, lambda=1
JiT	adamw	3	1e-05	0	sigma=0.05, pert.=1
RSF	adamw	5	1e-04	0	ρ=0.1

(b) random-20

Method	Opt.	Ep.	LR	WD	Method-specific HPs
Retrain-KD	adamw	10	3e-05	0	–
QUIT†	adamw	10	3e-05	0	tau=1, variant=logits_min
UniLogit†	adamw	3	3e-05	0	tau=1
UNDIAL†	adamw	10	1e-04	0	γ=6, tau=1
SalUn†	adamw	1	1e-04	0	mask=0.1
NegGrad	adamw	3	1e-05	0	–
RL	adamw	3	3e-05	0	–
TLN	adamw	3	1e-05	0	delta=0.5, lambda=0.5
JiT	adamw	5	3e-06	0	sigma=0.1, pert.=3
RSF	adamw	10	3e-05	0	ρ=0.1

Table 18: Selected hyperparameters for Oxford-IIIT Pets / ViT-S.

(a) random-10

Method	Opt.	Ep.	LR	WD	Method-specific HPs
Retrain-KD	adamw	10	1e-04	0	–
QUIT†	adamw	10	3e-05	0	tau=1, variant=logits_min
UniLogit†	adamw	3	3e-05	0	tau=1
UNDIAL†	adamw	10	3e-05	0	γ=14, tau=1
SalUn†	adamw	3	3e-05	0	mask=0.05
NegGrad	adamw	3	1e-05	0	–
RL	adamw	10	1e-04	0	–
TLN	adamw	5	3e-05	0	delta=0.5, lambda=0.5
JiT	adamw	5	3e-06	0	sigma=0.05, pert.=1
RSF	adamw	10	1e-04	0	ρ=0.35

(b) random-20

Method	Opt.	Ep.	LR	WD	Method-specific HPs
Retrain-KD	adamw	10	1e-04	0	–
QUIT†	adamw	5	3e-05	0	tau=1, variant=logits_min
UniLogit†	adamw	5	1e-04	0	tau=1
UNDIAL†	adamw	5	1e-04	0	γ=8, tau=1
SalUn†	adamw	1	3e-05	0	mask=0.1
NegGrad	adamw	10	1e-06	0	–
RL	adamw	5	3e-05	0	–
TLN	adamw	3	1e-04	0	delta=0.5, lambda=0.5
JiT	adamw	3	3e-06	0	sigma=0.05, pert.=1
RSF	adamw	10	1e-04	0	ρ=0.4

Table 19: Selected hyperparameters for Stanford Cars / ViT-S.

Method	Opt.	Ep.	LR	WD	Method-specific HPs
Retrain-KD	adamw	3	1e-04	0	–
QUIT†	adamw	5	1e-05	0	tau=1, variant=logits_min
UniLogit†	adamw	3	1e-05	0	tau=1
UNDIAL†	adamw	3	3e-05	0	γ=10, tau=1
SalUn†	adamw	1	1e-05	0	mask=0.3
NegGrad	adamw	1	3e-05	0	–
RL	adamw	1	1e-05	0	–
TLN	adam	10	1e-05	1e-05	delta=0, lambda=3
JiT	sgd	1	0.006	0	sigma=0.05, pert.=10
RSF	adamw	3	1e-04	0	ρ=0.35

Table 20: Selected hyperparameters for Tiny-ImageNet / ViT-S / random-10.

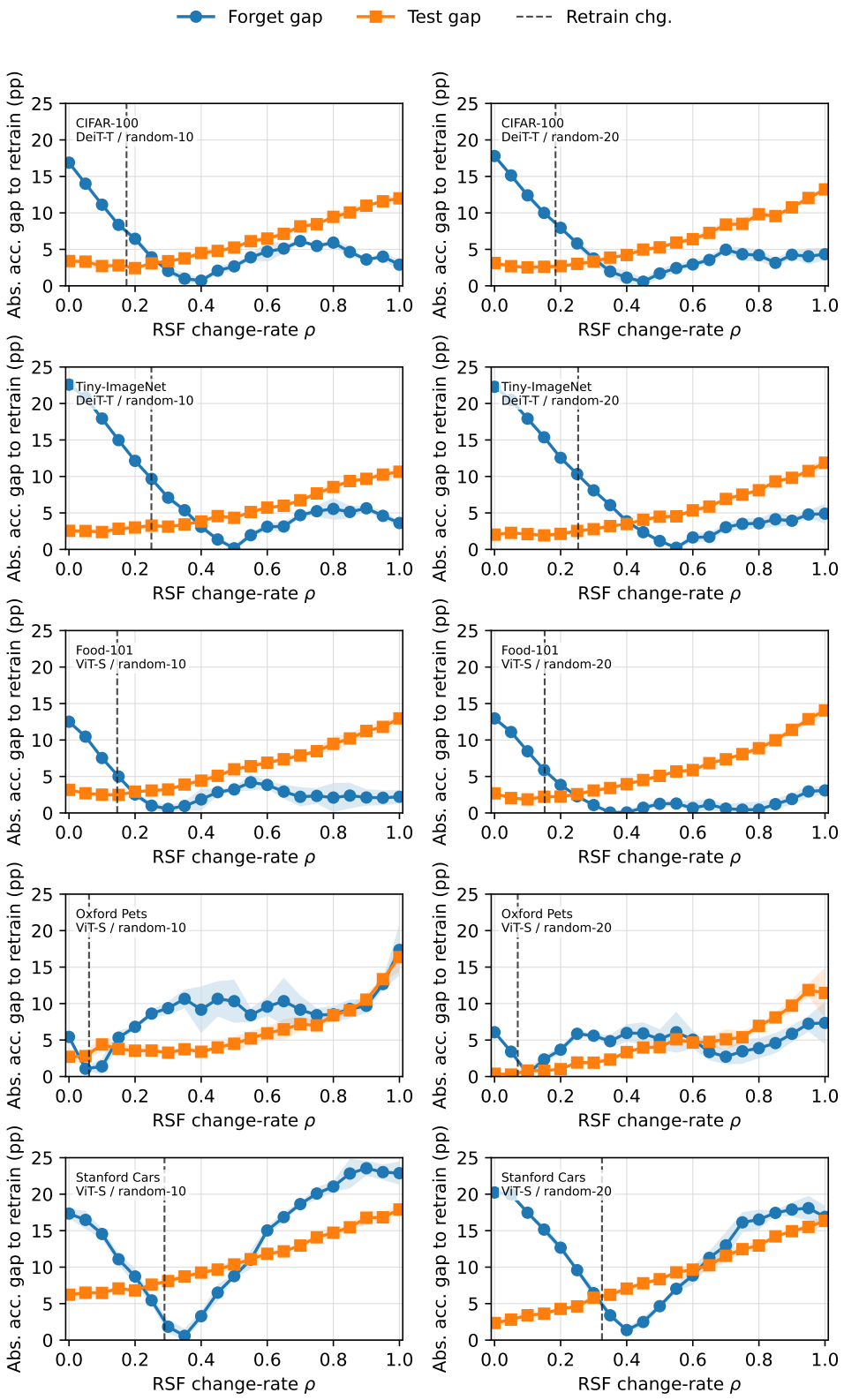


Figure 6: RSF sensitivity to the change budget ρ, reported as absolute gaps to retraining in forget accuracy (FA) and test accuracy (TA). Lower is better. Across settings, increasing ρ first sharply reduces the FA gap while the TA gap changes more gradually; after moderate budgets, additional interventions tend to enter an over-forgetting regime.

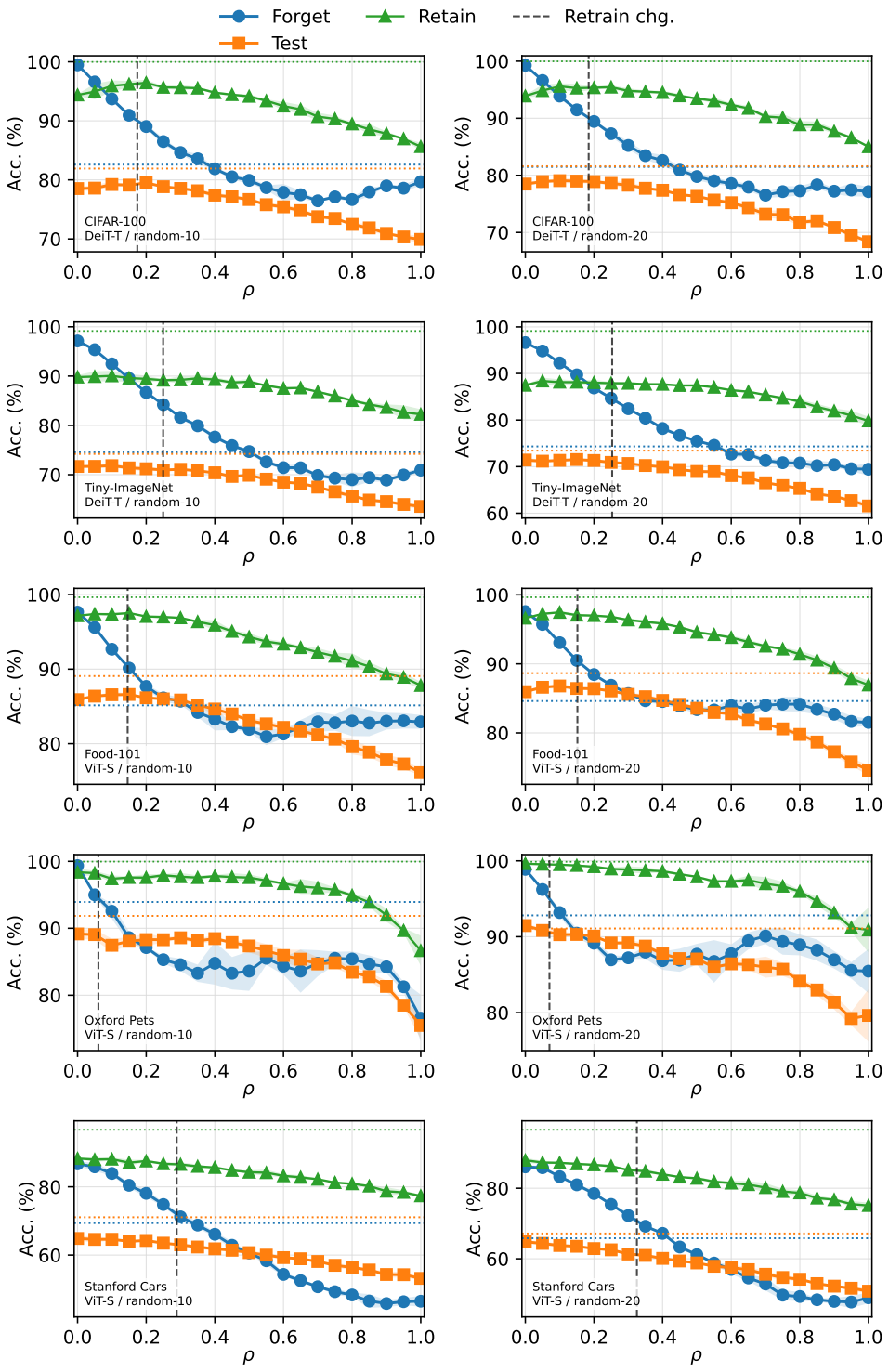


Figure 7: Full RSF accuracy sensitivity. The curves show forget, retain, and test accuracy as ρ varies, with dotted horizontal lines denoting the corresponding retrained references. Retain accuracy is comparatively stable over the useful middle range of ρ, while very large budgets can degrade both retain and test utility.

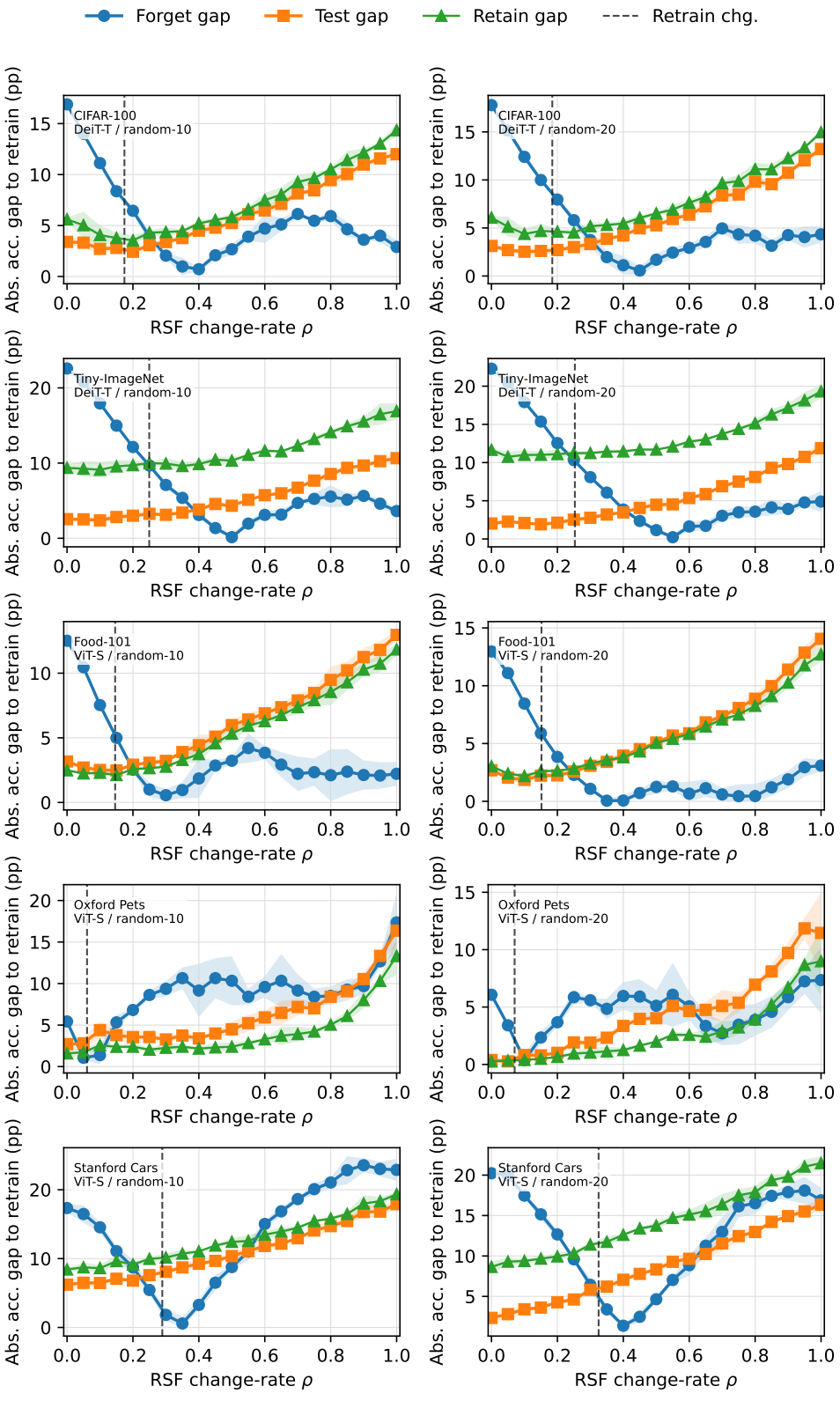


Figure 8: Full RSF accuracy-gap sensitivity, including forget, retain, and test gaps to retraining. This view makes explicit that ρ controls a forgetting–utility tradeoff rather than only forget-set behavior: budgets that best match forget accuracy can still increase retain/test gaps if pushed too far.

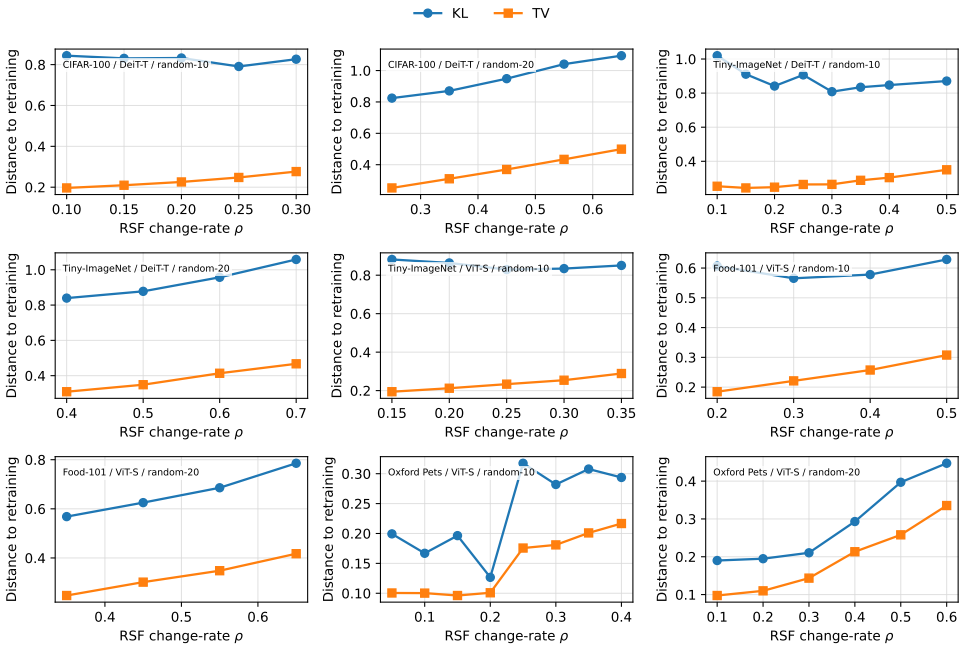


Figure 9: Distributional RSF sensitivity on forget samples. KL denotes $D_{\mathrm{KL}}(p^R \| p^U)$ and TV denotes total variation to the retrained reference. Distributional alignment usually improves when ρ moves away from under-intervention, but very large budgets are not uniformly beneficial, matching the accuracy tradeoff in Figures 6–8.