CURE: A UNIFIED FRAMEWORK FOR CLASS AND CONCEPT UNLEARNING VIA RETRAINING EMULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Driven by evolving data regulations and the need for trustworthy AI, machine unlearning (MU) addresses the critical challenge of efficiently removing undesirable knowledge from models without the prohibitive cost of retraining. However, existing MU methods have limitations in balancing the complete removal of target information and the degradation of performance on remaining data. In datasets with rich concept hierarchies, an additional trade-off exists between retaining knowledge of closely related concepts and that of more general, unrelated ones. We propose a Class and Concept Unlearning via Retraining Emulation (CURE) framework that preserves model performance by emulating retraining via knowledge distillation. We first formulate two unlearning strategies with mathematical justification: Guided Hard Relabeling (GHR) with cross-entropy and Guided Soft Relabeling (GSR) with Kullback-Leibler (KL) divergence. In datasets with extensive semantic hierarchies, we observe a key trade-off: GHR offers concentrated preservation for a small subset of closely related retain concepts, while GSR is more effective at preserving the wider set of dissimilar retain concepts. To unify these benefits, we introduce Guided Restricted Orthogonal Gradient Unlearning (GROGU) to optimize the update step by orthogonally combining gradients from two distillation-based objectives. Experiments on the image classification benchmarks CIFAR-10, CIFAR-100, and ImageNet, as well as on large language models (LLMs), show that our methods achieve superior target erasure while preserving accuracy on retained data, outperforming existing techniques.

1 INTRODUCTION

Growing concerns over model safety and data privacy have created a need to remove specific data from trained machine learning models. This requirement is supported by legal frameworks like the European Union's General Data Protection Regulation (GDPR), which establishes a "right to be forgotten" (Hoofnagle et al., 2019). As modern models are trained on increasingly large datasets (Morris et al., 2025; Villalobos et al., 2024), the simple solution of retraining the model from scratch is often too costly and time-consuming. To address this issue, the field of machine unlearning (MU) has emerged as a way to efficiently remove the influence of unwanted data while preserving the model's performance on all remaining knowledge (Bourtoule et al., 2021; Li et al., 2025b;a).

The initial motivation for machine unlearning stemmed from the principles of differential privacy (Dwork et al., 2014; Cao & Yang, 2015), where the primary goal was to create a model that was provably robust against privacy attacks seeking to infer information about individual training points (Dwork, 2006; Wasserman & Zhou, 2010). More recently, the unlearning problem has been formalized around a more practical objective: to efficiently approximate a "gold-standard" model that would have been produced by retraining from scratch on the dataset with the forget data removed (Bourtoule et al., 2021; Warnecke et al., 2021; Graves et al., 2021; Thudi et al., 2022; Jia et al., 2023; Warnecke et al., 2021; Golatkar et al., 2020; Becker & Liebig, 2022; Izzo et al., 2021; Ding et al., 2024). However, this objective presents a significant practical challenge, as this ideal retrained model is typically unavailable for direct comparison during the unlearning process.

A prominent category of machine unlearning techniques employs a fine-tuning approach to modify a pre-trained model (Warnecke et al., 2021; Golatkar et al., 2020). Within this paradigm, the unlearning process typically involves updating the model with respect to two objectives. To erase the target information, these methods often perform gradient ascent on the forget set to maximize its loss (Graves et al., 2021; Thudi et al., 2022) or, for language models, use techniques like Negative Preference Optimization (NPO) (Zhang et al., 2024). A significant drawback of this approach, however, is that gradient ascent is often a blunt instrument, causing catastrophic forgetting that severely degrades overall model utility. This highlights a key limitation: simply maximizing loss on the forget set fails to produce a model that accurately approximates one retrained from scratch. To mitigate this damage and preserve general knowledge, the "forget" step is often paired with a "retain" step, which performs standard gradient descent or KL-divergence minimization on a subset of the retain data. While many approaches are based on this combined approach with an improvement (Ko et al., 2024; Fan et al., 2023; Trippa et al., 2024), striking the right balance between the forget and retain objectives remains a significant challenge.

Knowledge distillation (KD), originally introduced by Hinton et al. (2015), provides a framework for transferring information from a larger "teacher" model to a smaller "student" model. While it was first developed as a strategy for model compression, it has since become a versatile tool with applications across various tasks (Gou et al., 2021; Mo et al., 2024; Salimans & Ho, 2022; Sanh et al., 2019; Sinha et al., 2023). In the context of machine unlearning, KD offers an attractive mechanism for guiding models to adapt their behavior while maintaining overall accuracy on non-forgotten knowledge (Dong et al., 2024; Yang, 2025; Vasilev et al., 2025; He et al., 2025; Chen et al., 2023; Zhou et al., 2025; Kim et al., 2024; Kim & Woo, 2022). While these studies demonstrate the promise of KD for effective forgetting and utility preservation, they do not explicitly address the inherent trade-off between maintaining accuracy on the relevant retained knowledge and preserving overall performance across all retain classes. Our work tackles this gap by directly modeling and analyzing this balance.

In this work, we propose CURE, a KD–based framework for machine unlearning. CURE consists of three complementary strategies: Guided Hard Relabeling (GHR), which maps each forget sample to the most likely non-forget label under the original model; Guided Soft Relabeling (GSR), which suppresses the forget-class logit and redistributes its probability mass across the remaining classes using temperature-scaled softmax; and Guided Restricted Orthogonal Gradient Unlearning (GROGU), which jointly optimizes both objectives. Empirically, we find that GHR better preserves accuracy on classes most related to the forget set, while GSR maintains higher overall retain performance. Among them, GROGU provides the best balance, particularly in the challenging case without access to retain data.

Overall, our contributions are:

- We introduce CURE, a unified KD–based framework for machine unlearning, with three strategies (GHR, GSR, GROGU) that capture complementary strengths.
- We highlight the trade-off between perserving relevant retain accuracy and overall retain accuracy, which prior KD-based unlearning methods have largely overlooked.
- We demonstrate that GROGU achieves state-of-the-art unlearning performance in the challenging setting without access to retain data.
- We additionally analyze the case where limited retain data are available and show that GSR can effectively leverage this signal through temperature scaling, with results included in the appendix.
- We conduct extensive experiments on CIFAR-10, CIFAR-100, and ImageNet for classification tasks, as well as autoregressive LLMs, comparing against strong unlearning baselines. We report comprehensive results with detailed analysis and visualizations, showing that CURE—especially GROGU—achieves state-of-the-art performance across diverse challenging forgetting scenarios.

2 Preliminaries

To illustrate our approach, we focus on the task of class-centric machine unlearning, where we introduce the relevant notions and provide a clear problem setup. Although we present the formulation in this setting, our solution is general and we empirically validate it on LLMs as well.

2.1 Problem Formulation.

We consider the supervised learning setting with training dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N \subseteq \mathcal{X} \times \mathcal{Y}$, containing N inputs $\mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^d$ and outputs $y_i \in \mathcal{Y} \subseteq \mathbb{R}^n$. For image classification, $\mathcal{Y} = \{1, \ldots, K\}$ specifies one of K classes. The model trained on the full dataset $\mathcal{D}$ is written as $f_{\theta_o} : \mathcal{X} \to \mathcal{Y}$, where θ_o are the original learned parameters. For machine unlearning, we distinguish between two disjoint subsets of the training data: the forget set $\mathcal{D}_f \subseteq \mathcal{D}$ containing samples that are asked to be erased, and the retain set $\mathcal{D}_r = \mathcal{D} \setminus \mathcal{D}_f$ containing all remaining samples. The goal is to update the model so that its behavior reflects training only on $\mathcal{D}_r$, as if $\mathcal{D}_f$ had never been seen.

Given an original model f_{θ_o}, an unlearning algorithm updates its parameters using $\mathcal{D}_f$ to obtain a new model f_θ, which should forget the information in $\mathcal{D}_f$ while retaining good performance on $\mathcal{D}_r$. As a reference, we define the gold-standard model f_{θ^*}, obtained by retraining from scratch on $\mathcal{D}_r$ alone. This retrained model is typically regarded as the ideal target for machine unlearning, and the goal is for f_θ to approximate f_{θ^*} as closely as possible. For clarity of exposition, we focus our problem formulation on the case where only $\mathcal{D}_f$ is available during unlearning, though in our experiments we also study settings where limited retain data can be incorporated by optimizing with a weighted retain loss.

2.2 Theoretical motivation

Inspired by Decoupled Knowledge Distillation (DKD) (Zhao et al., 2022) and DELETE (Zhou et al., 2025), we begin by analyzing the unlearning loss when formulated through KL divergence between a target distribution and the model's prediction. Since one-hot labels are a special case of probability distributions, this perspective allows us to connect the decomposed terms of the KL loss with three distinct roles in the unlearning setting: the forget class, the relevant retain class, and the remaining general retain classes.

Fix $u \in \{1, \ldots, K\}$ as the target forget class and fix a sample x in the forget set $\mathcal{D}_f$. We define $r(x) = \arg\max_{k \neq u}(f_{\theta_o}(x))_k$, the non-forget class with the highest logit under the original model f_{θ_o}. Intuitively, $r(x)$ captures the "most relevant" alternative class for x once its original label u is suppressed. We denote by $R = \{r(x) : x \in \mathcal{D}_f\}$ the set of all such relevant classes. Empirically, R is a non-empty set and much smaller than K.

Continuing with the same forget sample $x \in \mathcal{D}_f$ with original label u, we let $\mathbf{p} \in \mathbb{R}^K$ denote the target probability distribution and $\mathbf{q} \in \mathbb{R}^K$ the output distribution of the unlearning model f_θ. It is natual to define an unlearning objective as the KL divergence between these two distributions $\mathcal{L} = D_{\mathrm{KL}}(\mathbf{p} \,\|\, \mathbf{q})$. Following the notation of Zhou et al. (2025), we let p_i and q_i denote the i-th components of the target distribution $\mathbf{p}$ and model prediction $\mathbf{q}$, respectively. In particular, p_u and q_u correspond to the forget class u, with $p_{\setminus u} = \sum_{i \neq u} p_i$ and $q_{\setminus u} = \sum_{i \neq u} q_i$ denoting the aggregated probabilities of all other classes. For $i \neq u$, we define the normalized distributions $\tilde{\mathbf{p}}$ and $\tilde{\mathbf{q}}$ over the remaining $K-1$ classes, where $\tilde{p}_i = p_i / p_{\setminus u}$ and $\tilde{q}_i = q_i / q_{\setminus u}$. With this notation, the KL divergence between $\mathbf{p}$ and $\mathbf{q}$ can be re-expressed as

$$\mathcal{L} = p_u \log\left(\frac{p_u}{q_u}\right) + p_{\setminus u} \log\left(\frac{p_{\setminus u}}{q_{\setminus u}}\right) + p_{\setminus u}\tilde{p}_r \log\left(\frac{\tilde{p}_r}{\tilde{q}_r}\right) + p_{\setminus u} \sum_{i \neq u, r} \tilde{p}_i \log\left(\frac{\tilde{p}_i}{\tilde{q}_i}\right), \tag{1}$$

where, for notational simplicity, we use r instead of $r(x)$. We write $\mathbf{p}^{(b)} = [p_u, p_{\setminus u}]$ and $\mathbf{q}^{(b)} = [q_u, q_{\setminus u}]$ for the binary distributions over $\{u, \setminus u\}$. Then equation can be further written as

$$\mathcal{L} = D_{\mathrm{KL}}(\mathbf{p}^{(b)} \,\|\, \mathbf{q}^{(b)}) + p_{\setminus u}\tilde{p}_r \log\left(\frac{\tilde{p}_r}{\tilde{q}_r}\right) + p_{\setminus u} \sum_{i \neq u, r} \tilde{p}_i \log\left(\frac{\tilde{p}_i}{\tilde{q}_i}\right). \tag{2}$$

Ideally, the target distribution $\mathbf{p}$ would be given by the retrained model f_{θ^*} evaluated on x. However, since f_{θ^*} is not accessible during unlearning, we approximate $\mathbf{p}$ using the logits of the original model f_{θ_o}. Specifically, we set $\mathbf{p} = \mathrm{softmax}(f_{\theta_o}(x) - \infty \cdot \mathbf{e}^u)$, which ensures that $p_u = 0$ and the probability mass is redistributed across the remaining $K - 1$ classes.

To that end, the first term in equation (2) associates with forgetting the target class u; the second term linked to its most relevant class $r(x)$; and the third term covering the remaining retain classes. This decomposition naturally motivates different strategies. If we relabel x to $r(x)$, the KL loss reduces to forgetting class u while explicitly supervising only the relevant class $r(x)$. While this decomposition is written for a single instance x, the empirical risk minimized during training averages over all $x \in \mathcal{D}_f$. As a result, the supervision from the second term extends beyond individual $r(x)$ and effectively preserves the predictions for the small subset of classes R. Note that this strategy leaves a large general retain classes unsupervised since with relabeling $\tilde{p}_i = 0, \forall i \in \mathcal{Y}/\{u, r\}$.

On the other hand, if we use a distillation temperature softened probability distribution as the target, probability mass is redistributed across multiple classes rather than being concentrated entirely on $r(x)$, or R. In this case, the second and third terms in equation (2) combine into $p_{\setminus u} D_{\mathrm{KL}}(\tilde{\mathbf{p}} \,\|\, \tilde{\mathbf{q}})$, which provides supervision to all retain classes, with $\tilde{p}_i$ determining the relative emphasis across classes. Higher temperatures increase the entropy of $\tilde{\mathbf{p}}$, spreading supervision more uniformly across retain classes, whereas lower temperatures concentrate it on a few classes with larger $\tilde{p}_i$.

3 Method

3.1 Motivation

The decomposition in Section 2.2 shows that the KL divergence for a forget sample naturally splits into contributions from the forget class, the most relevant class, and the remaining retain classes. This observation suggests two complementary strategies: one that explicitly supervises the relevant class through relabeling, and another that redistributes probability mass across all retain classes via soft targets. We now introduce our proposed framework, CURE, which instantiates both strategies as Guided Hard Relabeling (GHR) and Guided Soft Relabeling (GSR), and further combines them into Guided Restricted Orthogonal Gradient Unlearning (GROGU).

3.2 Proposed Method

3.2.1 Guided Hard Relabel (GHR)

For each forget sample $x \in \mathcal{D}_f$ with original label u, we define the guided hard label as $r(x) = \arg\max_{k \neq u}(f_{\theta_o}(x))_k$. We then replace the target label u with $r(x)$ and optimize using the standard cross-entropy loss. The GHR objective is given by

$$\mathcal{L}_{\mathrm{GHR}}(\theta) = \frac{1}{|\mathcal{D}_f|} \sum_{x \in \mathcal{D}_f} \ell_{\mathrm{CE}}\big(f_\theta(x),\, r(x)\big),$$

where ℓ_{CE} denotes the cross-entropy loss. This strategy enforces forgetting of class u while preserving predictions on the small subset of relevant classes R.

3.2.2 Guided Soft Relabel (GSR)

Instead of relabeling to a single class, GSR constructs a soft target distribution by suppressing the forget-class logit in the original model and applying a temperature-scaled softmax. Specifically, for $x \in \mathcal{D}_f$ we define

$$\mathbf{p}_T(x) = \mathrm{softmax}\left(\frac{f_{\theta_o}(x) - \infty \cdot \mathbf{e}^u}{T}\right), \qquad \mathbf{q}_T(x;\theta) = \mathrm{softmax}\left(\frac{f_\theta(x)}{T}\right),$$

where $T > 0$ is a temperature hyperparameter. The GSR objective is then

$$\mathcal{L}_{\mathrm{GSR}}(\theta) = \frac{T^2}{|\mathcal{D}_f|} \sum_{x \in \mathcal{D}_f} D_{\mathrm{KL}}(\mathbf{p}_T(x) \,\|\, \mathbf{q}_T(x;\theta)).$$

This loss enforces zero probability on the forget class while distributing supervision across all retain classes, with the degree of uniformity controlled by T.

3.2.3 Guided Restricted Orthogonal Gradient Unlearning (GROGU)

GHR and GSR emphasize different aspects of supervision: GHR focuses on preserving accuracy for the small subset of relevant classes R, while GSR distributes supervision more broadly across all retain classes. GROGU combines these complementary signals using restricted orthogonal gradients (Ko et al., 2024). Let

$$\delta_H := \nabla_\theta \mathcal{L}_{\mathrm{GHR}}(\theta), \qquad \delta_S := \nabla_\theta \mathcal{L}_{\mathrm{GSR}}(\theta).$$

We then remove conflicting components by projecting each gradient onto the orthogonal complement of the other:

$$\mathrm{Proj}_b^{\perp}(a) := a - \frac{\langle a, b\rangle}{\|b\|_2^2 + \varepsilon}\, b,$$

with a small $\varepsilon > 0$ added for numerical stability. The GROGU update direction is then

$$\mathrm{Proj}_{\delta_S}^{\perp}(\delta_H) \;+\; \mathrm{Proj}_{\delta_H}^{\perp}(\delta_S).$$

This ensures that the complementary contributions of GHR and GSR are preserved without destructive interference.

4 Related Work

Several recent approaches have explored knowledge distillation for unlearning by modifying target distributions. UNDIAL (Dong et al., 2024), CE-U (Yang, 2025), and DELETE (Zhou et al., 2025) applies self-distillation after reducing the logit of the forget class by a fixed constant, but only evaluates the method with a default temperature of $T = 1$. We find that adjusting the temperature is crucial for managing the trade-off between forgetting strength and retention on general retain classes. Building on the idea of logit adjustment for unlearning, Unilogit (Vasilev et al., 2025) assigns a uniform probability mass to the forget token while leaving the other logits unchanged, which corresponds to the limiting case of an infinitely high distillation temperature. Although this removes the need for tuning, it often overshoots, leading to degraded performance on retain classes. By contrast, our GSR and GROGU explicitly incorporates temperature scaling, and we provide practical guidance for choosing effective values across datasets of varying complexity in Appendix A.3.

Other works have also incorporated knowledge distillation but with narrower applicability. LAU (Kim et al., 2024) combines adversarial perturbations at the classification layer with distillation from a frozen teacher to maintain decision boundaries, enabling efficient unlearning for image classifiers. However, this method does not extend naturally to other architectures, whereas our framework applies to both classification tasks and autoregressive language models by masking logits associated with forget concepts. Two-stage retraining (Kim & Woo, 2022) improves efficiency over training from scratch by first neutralizing the model on the forget set with contrastive labels, then retraining on retain data using distillation from the original model. While effective, this approach fundamentally depends on access to retain data. In contrast, CURE is applicable in both scenarios: we focus our main results on the more challenging no-retain-data case, and include additional experiments in the Appendix A.5-A.7 showing that it can also leverage small amounts of retain data when available.

5 Experiment and analysis

5.1 Experiment Setup

Datasets. Our class unlearning experiments are conducted on three standard image classification benchmarks: CIFAR-10, CIFAR-100 Krizhevsky (2009), and a pre-packaged version of ImageNet-1K Russakovsky et al. (2015) from the Hugging Face Hub. For all datasets, we use their standard training sets for model training. We randomly split the original test set (for CIFAR-10/100) and the original validation set (for ImageNet) in half, using one half as our validation set and the other as our final test set. Further details on all datasets are provided in Appendix A.1.

Image Classification Evaluation Metrics. We evaluate model performance using four key metrics. 1) Test Accuracy: The overall accuracy on the entire test set. 2) Test Retain Accuracy: The accuracy on the subset of test data corresponding to all non-forgotten classes 3) Test Forget Accuracy: The accuracy on the subset of test data belonging to the class that was unlearned. A perfect unlearning method should maximize Test Retain Accuracy while driving Test Forget Accuracy to zero. Finally, to analyze performance degradation at a more in-depth level, we measure the 4) Relative Accuracy Change, which is the per-class percentage change in accuracy on each retained class compared to a model retrained from scratch.

LLM Evaluation Metrics. For the concept unlearning task on LLMs, our evaluation protocol assesses both unlearning efficacy and model utility. To quantify the efficacy of forgetting, we use three metrics. 1) Extraction Strength (ES) measures the intensity of memorization by determining the minimum prompt prefix required to reconstruct a forgotten fact; a lower ES indicates more successful erasure. 2) QA Probability directly measures the model's confidence in generating the forgotten information in a question-answering context. 3) QA ROUGE assesses the textual overlap between the model's output and the ground-truth forgotten answer. For both QA Probability and QA ROUGE, lower scores signify more effective unlearning. To measure the impact on the model's general capabilities, we report a single, composite Utility score. This metric captures the model's retained performance on a broad range of tasks beyond the unlearning target, ensuring that unlearning does not cause catastrophic degradation of its general knowledge. A higher Utility score is desirable, indicating the model remains useful after the unlearning process.

Baseline Models. To demonstrate the effectiveness of our CURE framework, we conduct a comprehensive evaluation against a wide spectrum of existing unlearning methods across both image and language domains. For image classification, we include gradient ascent (GradAsc), random label (RandLabel), finetunr (Finetune), AGE (Bui et al., 2025)), UNDIAL (Dong et al., 2024)). We evaluate these methods in a challenging forget-data-only scenario. In addition, for the retain-data-aided setting, we include gradient ascent descent (GradASc Descent), restricted gradient descent (GradRest Descent), AGE Descent, UNDIAL Descent. For the LLM concept unlearning task, we benchmark CURE against naive approach GradAsc, as well as recent state-of-the-art methods GradDiff (Yao et al., 2024), NPO (Zhang et al., 2024), and SimNPO Fan et al. (2025), and UNDIAL (Dong et al., 2024)). A complete description of all baseline models is provided in Appendix A.2.

Implementation Details. Our experiments utilize ResNet-18 for CIFAR-10/100 and ResNet-50 for ImageNet. To ensure robustness on CIFAR-10/100, we train 30 distinct Original models, from which each unlearning method is run for 30 trials. For the Retrain gold standard, we also train 30 models from scratch on the training set with the respective forget-class data removed. For experiments incorporating retain data, we adopt a computationally efficient approach that also mimics realistic scenarios where access to the full retain set is impractical. This subset corresponds to approximately 34% of the available retain data for CIFAR-10, 62% for CIFAR-100, and 0.55% for ImageNet. This is achieved by cycling through the much smaller forget loader while drawing corresponding batches from the retain loader. Due to the cost of ImageNet training, we use a single Original and Retrain model, but still run each unlearning method for 30 trials to account for stochasticity in the unlearning process itself. More details about the hyperparameters of baseline models and CURE methods can be found in Appendix A.3.

5.2 Empirical results and analysis

We present a comprehensive evaluation of our CURE framework, beginning with the most challenging and practical scenario: unlearning with access only to the forget data.

Overall Performance. The overall performance of CURE and baseline methods is summarized in Table 1. Across all datasets and unlearning scenarios, our proposed methods demonstrate a clear advantage. Naive baselines like GradAsc and AGE consistently result in catastrophic forgetting of retain information or fail to effectively erase the target class. In contrast, our CURE methods successfully drive forget accuracy to near-zero while preserving high retain accuracy. Most notably, GROGU (or its T=1 variant, GROGU T1) consistently emerges as the top-performing method, achieving the highest retain accuracy among all methods. For instance, in the challenging ImageNet experiment for forgetting class 108 sea anemone, GROGU is the only method that effectively erases the target knowledge while maintaining high utility. Furthermore, the consistently low standard deviation of GROGU across all experiments highlights its stable and reliable unlearning behavior.

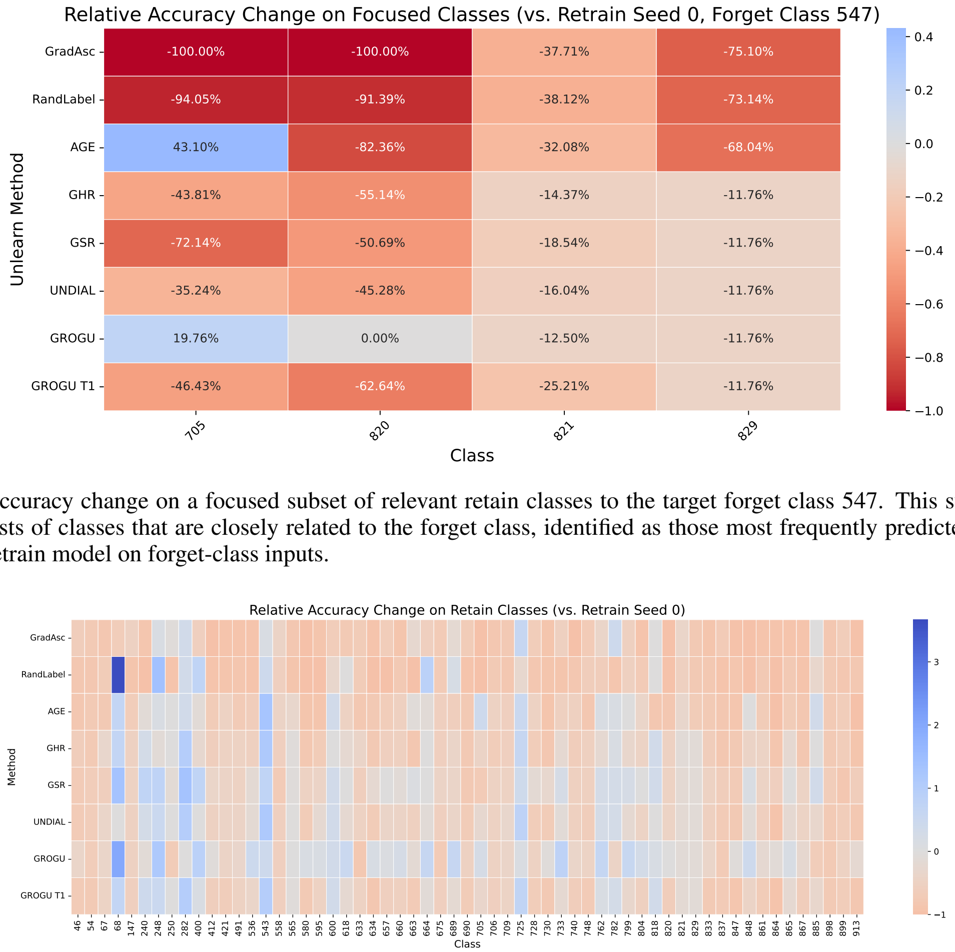
Table 1: Summary of performance for forget-data-only methods, representing the most challenging unlearning scenario. We report results for three scenarios (A, B, C) for each dataset, corresponding to forgetting classes $\{1, 3, 8\}$ for CIFAR-10, $\{16, 54, 81\}$ for CIFAR-100, and $\{108, 547, 892\}$ for ImageNet. For each scenario, we report the mean test retain and forget accuracy over 30 random seeds. The best-performing approximate unlearning method is highlighted in bold, selected by first meeting a dataset-specific forget accuracy threshold ($\leq$ 0.003 for CIFAR-10, $\leq$ 0.010 for CIFAR-100, and $\leq$ 0.001 for ImageNet) and then having the highest retain accuracy.

Method	**Scenario A**		**Scenario B**		**Scenario C**	
	Retain	**Forget**	**Retain**	**Forget**	**Retain**	**Forget**
CIFAR-10						
Original	0.926 ± 0.003	0.961 ± 0.006	0.926 ± 0.003	0.961 ± 0.006	0.926 ± 0.003	0.961 ± 0.006
Retrain	0.929 ± 0.003	0.000 ± 0.000	0.949 ± 0.002	0.000 ± 0.000	0.929 ± 0.002	0.000 ± 0.000
GradAsc	0.152 ± 0.020	0.000 ± 0.000	0.146 ± 0.019	0.000 ± 0.000	0.149 ± 0.018	0.000 ± 0.000
RandLabel	0.768 ± 0.087	0.044 ± 0.033	0.645 ± 0.084	0.082 ± 0.020	0.663 ± 0.108	0.038 ± 0.023
Finetune	0.928 ± 0.002	0.119 ± 0.133	0.949 ± 0.002	0.023 ± 0.022	0.927 ± 0.002	0.174 ± 0.139
AGE	0.209 ± 0.208	0.061 ± 0.094	0.123 ± 0.033	0.132 ± 0.120	0.132 ± 0.019	0.021 ± 0.047
GHR	0.862 ± 0.020	0.000 ± 0.001	0.923 ± 0.015	0.000 ± 0.001	0.857 ± 0.018	0.000 ± 0.000
GSR	0.854 ± 0.020	0.000 ± 0.001	0.886 ± 0.041	0.001 ± 0.002	0.614 ± 0.114	0.000 ± 0.001
UNDIAL	0.817 ± 0.045	0.000 ± 0.000	0.930 ± 0.012	0.000 ± 0.000	0.641 ± 0.070	0.002 ± 0.003
GROGU	**0.893 ± 0.007**	**0.000 ± 0.001**	0.929 ± 0.015	0.000 ± 0.000	0.851 ± 0.023	0.000 ± 0.000
GROGU T1	0.874 ± 0.038	0.002 ± 0.014	**0.940 ± 0.004**	**0.000 ± 0.000**	**0.873 ± 0.018**	**0.000 ± 0.000**
CIFAR-100						
Original	0.728 ± 0.004	0.746 ± 0.042	0.728 ± 0.004	0.746 ± 0.042	0.728 ± 0.004	0.746 ± 0.042
Retrain	0.731 ± 0.004	0.000 ± 0.000	0.729 ± 0.004	0.000 ± 0.000	0.732 ± 0.004	0.000 ± 0.000
GradAsc	0.564 ± 0.032	0.018 ± 0.022	0.494 ± 0.047	0.012 ± 0.015	0.518 ± 0.050	0.011 ± 0.022
RandLabel	0.572 ± 0.014	0.033 ± 0.026	0.588 ± 0.014	0.023 ± 0.026	0.579 ± 0.018	0.038 ± 0.034
Finetune	0.727 ± 0.005	0.488 ± 0.050	0.726 ± 0.005	0.547 ± 0.049	0.727 ± 0.005	0.275 ± 0.075
AGE	0.524 ± 0.034	0.031 ± 0.032	0.537 ± 0.027	0.021 ± 0.018	0.557 ± 0.027	0.028 ± 0.038
GHR	0.573 ± 0.038	0.015 ± 0.023	0.542 ± 0.049	0.015 ± 0.025	0.592 ± 0.030	0.009 ± 0.016
GSR	0.584 ± 0.018	0.014 ± 0.021	0.590 ± 0.018	0.010 ± 0.018	0.582 ± 0.033	0.015 ± 0.024
UNDIAL	0.551 ± 0.039	0.027 ± 0.026	0.577 ± 0.021	0.013 ± 0.015	0.579 ± 0.028	0.030 ± 0.030
GROGU	**0.598 ± 0.011**	**0.008 ± 0.022**	**0.611 ± 0.007**	**0.005 ± 0.012**	**0.600 ± 0.008**	**0.004 ± 0.008**
GROGU T1	0.591 ± 0.011	0.022 ± 0.022	0.605 ± 0.007	0.021 ± 0.018	0.595 ± 0.012	0.033 ± 0.030
ImageNet						
Original	0.708	0.792	0.708	0.792	0.708	0.792
Retrain	0.703	0.000	0.706	0.000	0.696	0.000
GradAsc	0.472 ± 0.005	0.000 ± 0.000	0.599 ± 0.002	0.000 ± 0.000	0.647 ± 0.001	0.000 ± 0.000
RandLabel	0.517 ± 0.015	0.000 ± 0.000	0.596 ± 0.014	0.004 ± 0.012	0.621 ± 0.003	0.000 ± 0.000
Finetune	0.700 ± 0.001	0.438 ± 0.096	0.700 ± 0.001	0.362 ± 0.037	0.701 ± 0.002	0.083 ± 0.000
AGE	0.555 ± 0.003	0.000 ± 0.000	0.501 ± 0.010	0.000 ± 0.000	0.611 ± 0.002	0.000 ± 0.000
GHR	0.589 ± 0.002	0.000 ± 0.000	0.662 ± 0.001	0.154 ± 0.000	0.639 ± 0.011	0.000 ± 0.000
GSR	0.619 ± 0.002	0.000 ± 0.000	0.650 ± 0.015	0.015 ± 0.020	0.670 ± 0.001	0.000 ± 0.000
UNDIAL	0.603 ± 0.002	0.000 ± 0.000	0.666 ± 0.001	0.077 ± 0.000	0.641 ± 0.001	0.000 ± 0.000
GROGU	**0.623 ± 0.001**	**0.000 ± 0.000**	**0.655 ± 0.001**	**0.000 ± 0.000**	**0.684 ± 0.001**	**0.000 ± 0.000**
GROGU T1	0.574 ± 0.003	0.000 ± 0.000	0.626 ± 0.004	0.000 ± 0.000	0.644 ± 0.002	0.000 ± 0.000

The Role of Temperature in Forgetting. An interesting secondary observation from Table 1 concerns the role of distillation temperature in the forget-data-only setting. By comparing GSR to its $T = 1$ counterpart, UNDIAL, we see that UNDIAL exhibits a higher forget accuracy than GSR in the majority of cases. This suggests that a higher distillation temperature not only helps in knowledge distillation for preserving retain accuracy but also plays a role in facilitating more effective erasure of the forget class.

The GHR/GSR Trade-off and GROGU's Unification. While overall metrics establish the superiority of our framework, an in-depth analysis of per-class accuracy reveals the key scientific insight that motivates our final method, GROGU. The accuracy drop heatmaps in Fig. 2- 1 visualize the performance change on each retain class relative to a fully retrained model. These plots clearly show that all our CURE methods cause significantly less performance degradation than baselines like GradAsc, RandLabel, and AGE.

The more critical finding, however, is the distinct trade-off between our simpler guided relabeling methods, GHR and GSR. This is best illustrated on ImageNet using Fig. 1a, which shows a focused



(a) Accuracy change on a focused subset of relevant retain classes to the target forget class 547. This subset consists of classes that are closely related to the forget class, identified as those most frequently predicted by the retrain model on forget-class inputs.

(b) Relative accuracy change on a general subset of ImageNet retain classes. To visualize performance on diverse classes, this heatmap displays the 60 retain classes with the highest performance variance across the unlearning methods

Figure 1: Relative accuracy change on the ImageNet retain set for forget-data-only methods after unlearning class 547 electric locomotive. **(a)** Performance on a focused subset of semantically related classes. **(b)** Performance on a broad subset of general retain classes. In both heatmaps, red indicates an accuracy drop, gray indicates negligible change, and blue indicates a gain relative to the retrained model.

subset of closely relevant classes, and Fig. 1b, which shows a broad set of general retain classes.In Fig. 1a, GHR outperforms GSR, showing lower accuracy drops on each of the four relevant retain classes. In particular, GSR struggles significantly, exhibiting a severe accuracy drop of -72.14% on class 705 (passenger car). This class is the most relevant to the target forget class 547 (electric locomotive), as it is the class that the retrained model most frequently misclassifies forget-class images as. On the other hand, in Fig. 1b, a comparison of accuracy drops across a broad set of general retain classes reveals that GSR has more gray or blue cells while GHR has more orange cells, indicating that GHR causes widespread, moderate damage in this general setting.

This is precisely the conflict that GROGU is designed to resolve. By orthogonally combining the gradients from both objectives, GROGU inherits the best of both worlds. As shown in the bottom row of both heatmaps, it achieves strong performance on relevant classes, even exceeding GHR, while also maintaining high accuracy on general classes, similar to GSR. It thus provides a unified solution that robustly preserves performance across the entire spectrum of retain classes. We observe the same pattern on ImageNet when forgetting class 108 (sea anemone) and class 892 (wall clock). As shown in Appendix A.7 Fig. 39-42. This effect is also observable on simple structure dataset CIFAR-10 in Fig. 2, which shows the results for unlearning class 1 (automobile). In this scenario,

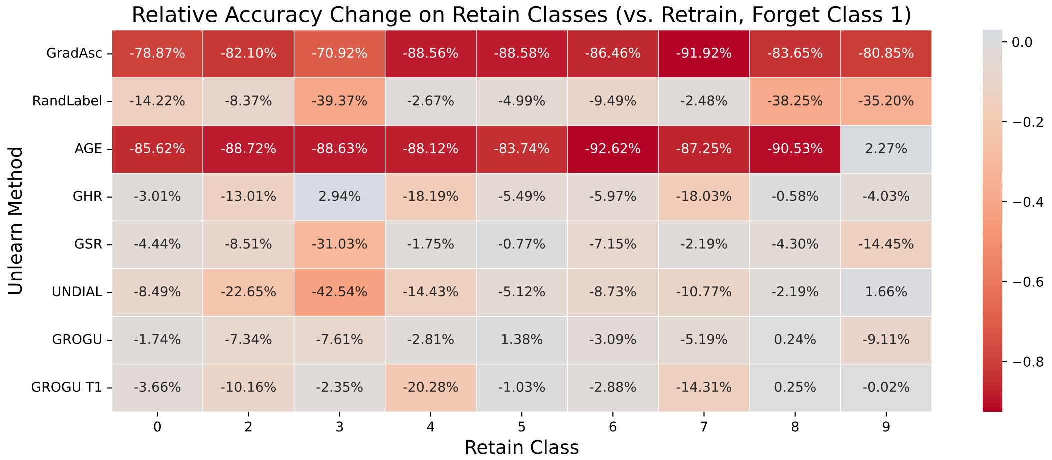


Figure 2: Relative accuracy change for each unlearning method (rows) across the nine CIFAR-10 retain classes (columns) in the forget-data-only setting. The color of each cell represents the performance change relative to a fully retrained model: red indicates an accuracy drop, gray indicates negligible change, and blue indicates an accuracy gain.

GROGU's accuracy drop on the relevant class 9, truck, is a modest -9.11% compared to GSR's -14.45%, while it simultaneously causes less damage than GHR across the other retain classes.

Connection to the temperature. The trade-off between relevant and general class preservation is directly linked to the distillation temperature. Low-temperature methods like UNDIAL ($T = 1$) behave similarly to the hard-label GHR, focusing preservation on a few relevant classes at the expense of general ones, as seen in Fig. 2- 1. Higher temperatures generalize preservation but weaken the focus on relevant classes. GROGU allows it to operate at a lower, more focused temperature than GSR (e.g., $T = 15$ for GROGU vs. $T = 70$ for GSR on ImageNet) without sacrificing general-class performance. This demonstrates that the restricted orthogonal gradient is the key to achieving both focused and broad knowledge preservation simultaneously.

Additional Classification Analyses. Our comprehensive experimental results, including several supplementary analyses, are detailed in the appendix. This includes the full performance tables and accuracy drop heatmaps for all unlearning scenarios on CIFAR-10 (Appendix A.5), CIFAR-100 (Appendix A.6), and ImageNet (Appendix A.7). Crucially, these sections also contain a complete evaluation of methods that incorporate retain data. Furthermore, we provide prediction count distributions for all datasets to demonstrate that our CURE methods closely emulate the prediction patterns of the retrained model, outperforming the baselines in alignment. Finally, for the CIFAR-10 case (Appendix A.5), we include an analysis of the gradients with respect to the input pixels. We find that models unlearned with AGE, UNDIAL, and our CURE framework exhibit a significantly smaller Euclidean distance to the retrain model's gradients compared to naive baselines like GradAsc, RandLabel, GradAsc Descent, GradRest Descent, indicating a closer alignment in input sensitivity.

Large Language Models. To demonstrate the versatility of our framework, we apply GROGU to a concept unlearning task on Large Language Models, with results summarized in Table 2. We perform this evaluation on TOFU (Maini et al., 2024) to unlearn synthetic data about fictitious authors. To apply GROGU to this problem, we simply treat the names of the authors belonging to the forget set as forget classes.

While simple baselines like GradAsc can achieve near-perfect unlearning efficacy, showing an ES of 0.033, they do so at the cost of completely destroying the model's general capabilities, resulting in a Utility score of zero. More advanced baselines like UNDIAL, GradDiff and NPO improve upon this but still exhibit a large drop in utility. While SimNPO has higher Utility, it also has higher ES, QA Probability and QA ROUGE scores. In contrast, our GROGU variants achieve a higher Utility score while maintaining competitively low ES, QA Probability and QA ROUGE scores.

Table 2: Performance of GROGU and baseline methods on the LLM concept unlearning task. Lower is better for efficacy metrics (ES, QA Prob., QA ROUGE), while higher is better for Utility. Our GROGU variants achieve the best balance, maintaining high Utility while effectively erasing the target concept.

Method	**ES**	**QA Prob.**	**QA ROUGE**	**Utility**
GradAsc	0.033	2.6e-6	0.047	0.000
UNDIAL	0.271	0.490	0.531	0.572
GradDiff	0.082	0.058	0.354	0.442
NPO	0.096	0.213	0.198	0.438
SimNPO	0.560	0.843	0.734	0.598
CURE (GROGU, T=1, w=0.5)	0.241	0.477	0.471	0.607
CURE (GROGU, T=5, w=0.5)	0.243	0.547	0.480	**0.608**
CURE (GROGU, T=5, w=10)	0.121	0.421	0.390	0.588

This demonstrates GROGU's ability to effectively erase targeted concepts from LLMs without the catastrophic degradation of model utility.

6 Conclusion

We presented CURE, a knowledge-distillation–based framework for machine unlearning designed to operate effectively even in the challenging setting where no retain data are available. Through theoretical analysis, we revealed that unlearning involves distinct components tied to the forget class, its closely related retain classes, and the broader set of general retain classes. This perspective allowed us to identify and formalize a previously overlooked trade-off: preserving accuracy on relevant retain classes versus maintaining performance on general retain classes.

Building on this insight, we developed Guided Hard Relabeling (GHR) to emphasize relevant classes, Guided Soft Relabeling (GSR) to preserve general retain classes, and Guided Restricted Orthogonal Gradient Unlearning (GROGU) to reconcile the two objectives. Extensive experiments on CIFAR-10, CIFAR-100, ImageNet, and autoregressive LLMs demonstrate that CURE, and especially GROGU, achieves state-of-the-art forgetting while maintaining balanced retention. By directly addressing the retain-free setting and illuminating the nuanced trade-off in retention, this work provides both practical algorithms and conceptual clarity for advancing machine unlearning.

Author Contributions

If you'd like to, you may include a section for author contributions as is done in many journals. This is optional and at the discretion of the authors.

Acknowledgments

Use unnumbered third level headings for the acknowledgments. All acknowledgments, including those to funding agencies, go at the end of the paper.

References

Alexander Becker and Thomas Liebig. Evaluating machine unlearning via epistemic uncertainty. *arXiv preprint arXiv:2208.10836*, 2022.

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

Anh Bui, Trang Vu, Long Vuong, Trung Le, Paul Montague, Tamas Abraham, Junae Kim, and Dinh Phung. Fantastic targets for concept erasure in diffusion models and where to find them. *arXiv preprint arXiv:2501.18950*, 2025.

Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE symposium on security and privacy*, pp. 463–480. IEEE, 2015.

Min Chen, Weizhuo Gao, Gaoyang Liu, Kai Peng, and Chen Wang. Boundary unlearning, 2023. URL `https://arxiv.org/abs/2303.11570`.

Meng Ding, Rohan Sharma, Changyou Chen, Jinhui Xu, and Kaiyi Ji. Understanding fine-tuning in approximate unlearning: A theoretical perspective. *arXiv preprint arXiv:2410.03833*, 2024.

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. *arXiv preprint arXiv:2402.10052*, 2024.

Cynthia Dwork. Differential privacy. In *International colloquium on automata, languages, and programming*, pp. 1–12. Springer, 2006.

Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and trends® in theoretical computer science*, 9(3–4):211–407, 2014.

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

Chongyu Fan, Jiancheng Liu, Licong Lin, Jinghan Jia, Ruiqi Zhang, Song Mei, and Sijia Liu. Simplicity prevails: Rethinking negative preference optimization for llm unlearning. *URL https://arxiv. org/abs/2410.07163*, 2025.

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9304–9312, 2020.

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

Laura Graves, Vineel Nagisetty, and Vijay Ganesh. Amnesiac machine learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 11516–11524, 2021.

Zhengbao He, Tao Li, Xinwen Cheng, Zhehao Huang, and Xiaolin Huang. Towards natural machine unlearning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

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

Chris Jay Hoofnagle, Bart Van Der Sloot, and Frederik Zuiderveen Borgesius. The european union general data protection regulation: what it is and what it means. *Information & Communications Technology Law*, 28(1):65–98, 2019.

Zachary Izzo, Mary Anne Smart, Kamalika Chaudhuri, and James Zou. Approximate data deletion from machine learning models. In *International conference on artificial intelligence and statistics*, pp. 2008–2016. PMLR, 2021.

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.

Hyunjune Kim, Sangyong Lee, and Simon S Woo. Layer attack unlearning: Fast and accurate machine unlearning via layer level attack and knowledge distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 21241–21248, 2024.

Junyaup Kim and Simon S Woo. Efficient two-stage model retraining for machine unlearning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4361–4369, 2022.

Myeongseob Ko, Henry Li, Zhun Wang, Jonathan Patsenker, Jiachen Tianhao Wang, Qinbin Li, Ming Jin, Dawn Song, and Ruoxi Jia. Boosting alignment for post-unlearning text-to-image generative models. *Advances in Neural Information Processing Systems*, 37:85131–85154, 2024.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL `https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf`.

Chunxiao Li, Haipeng Jiang, Jiankang Chen, Yu Zhao, Shuxuan Fu, Fangming Jing, and Yu Guo. An overview of machine unlearning. *High-Confidence Computing*, 5(2):100254, 2025a.

Na Li, Chunyi Zhou, Yansong Gao, Hui Chen, Zhi Zhang, Boyu Kuang, and Anmin Fu. Machine unlearning: Taxonomy, metrics, applications, challenges, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*, 2025b.

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

Qijie Mo, Yipeng Gao, Shenghao Fu, Junkai Yan, Ancong Wu, and Wei-Shi Zheng. Bridge past and future: Overcoming information asymmetry in incremental object detection. In *European Conference on Computer Vision*, pp. 463–480. Springer, 2024.

John X Morris, Chawin Sitawarin, Chuan Guo, Narine Kokhlikyan, G Edward Suh, Alexander M Rush, Kamalika Chaudhuri, and Saeed Mahloujifar. How much do language models memorize? *arXiv preprint arXiv:2505.24832*, 2025.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

Yash Sinha, Murari Mandal, and Mohan Kankanhalli. Distill to delete: Unlearning in graph networks with knowledge distillation. *arXiv preprint arXiv:2309.16173*, 2023.

Anvith Thudi, Gabriel Deza, Varun Chandrasekaran, and Nicolas Papernot. Unrolling sgd: Understanding factors influencing machine unlearning. In *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*, pp. 303–319. IEEE, 2022.

Daniel Trippa, Cesare Campagnano, Maria Sofia Bucarelli, Gabriele Tolomei, and Fabrizio Silvestri. τ: Gradient-based and task-agnostic machine unlearning. *CoRR*, 2024.

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. *arXiv preprint arXiv:2505.06027*, 2025.

Pablo Villalobos, Anson Ho, Jaime Sevilla, Tamay Besiroglu, Lennart Heim, and Marius Hobbhahn. Position: Will we run out of data? limits of llm scaling based on human-generated data. In *Forty-first International Conference on Machine Learning*, 2024.

Alexander Warnecke, Lukas Pirch, Christian Wressnegger, and Konrad Rieck. Machine unlearning of features and labels. *arXiv preprint arXiv:2108.11577*, 2021.

Larry Wasserman and Shuheng Zhou. A statistical framework for differential privacy. *Journal of the American Statistical Association*, 105(489):375–389, 2010.

Bo Yang. Ce-u: Cross entropy unlearning. *arXiv preprint arXiv:2503.01224*, 2025.

Yuanshun Yao, Xiaojun Xu, and Yang Liu. Large language model unlearning. *Advances in Neural Information Processing Systems*, 37:105425–105475, 2024.

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

Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. Decoupled knowledge distillation. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pp. 11953–11962, 2022.

Yu Zhou, Dian Zheng, Qijie Mo, Renjie Lu, Kun-Yu Lin, and Wei-Shi Zheng. Decoupled distillation to erase: A general unlearning method for any class-centric tasks. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 20350–20359, 2025.

A Appendix

A.1 Datasets Details

Our experiments on class unlearning are conducted on three standard image classification benchmarks of increasing complexity: CIFAR-10, CIFAR-100, and ImageNet.

CIFAR-10 and CIFAR-100. The CIFAR-10 and CIFAR-100 datasets (Krizhevsky, 2009) are collections of 32x32 pixel color images, loaded directly via the `torchvision.datasets` library. CIFAR-10 consists of 60,000 images across 10 classes, while CIFAR-100 contains the same number of images distributed across 100 classes. For both datasets, we utilize the standard training set of 50,000 images. We randomly partition the original 10,000-image test set into two equal, non-overlapping halves: a 5,000-image validation set and a 5,000-image test set.

ImageNet. For our large-scale experiments, we use a version of the ILSVRC 2012 dataset (ImageNet-1K) (Russakovsky et al., 2015). Specifically, we utilize the prepackaged version available on the Hugging Face Hub under the repository name `benjamin-paine/imagenet-1k-128x128`, which is provided in the Parquet file format for efficient data loading. This version contains approximately 1.28 million training images, 50,000 validation images, and 100,000 test images. As the official test set labels are not publicly available, we use the standard training set for model training and create our own validation and test sets from the original 50,000-image validation set. We randomly partition it into two equal, non-overlapping halves: a 25,000-image validation set and a 25,000-image test set for final evaluation. As a final preprocessing step, all images are resized from their original 128x128 resolution to 224x224.

A.2 Baseline models details.

To evaluate the performance of our CURE framework, we compare it against a comprehensive set of existing unlearning methods. These methods are grouped into two categories based on their unlearning objective.

Image Classification Baselines. For our experiments on CIFAR-10, CIFAR-100, and ImageNet, we implement several established baselines. The first group uses only the forget data and includes: gradient ascent (GradAsc), which maximizes the loss on the forget set; Random Label (RandLabel), which fine-tunes on the forget set with shuffled labels; Finetune, which simply continues training on a random subset of retain data; UNDIAL (Dong et al., 2024), a knowledge distillation method using a temperature of $T = 1$ and therefore a special case of GSR; and AGE (Bui et al., 2025). AGE is a redirection method that maps forget class samples to a different class label which is closely related to the forget class. To ensure the redirection is meaningful, the target class for each unlearning task is determined empirically: we select the class that the original, pre-trained model most frequently misclassifies the forget-class images as. Following this procedure, our experiments on CIFAR-10 used the mappings of automobile (class 1) to truck (9), cat (3) to dog (5), and ship (8) to airplane (0). For CIFAR-100, we used the mappings of can (16) to lamp (40), orchid (54) to tulip (92), and streetcar (81) to bus (13). On ImageNet, the mappings were sea anemone (108) to sea snake (65), electric locomotive (547) to passenger car (705), and wall clock (892) to analog clock (409). The second group incorporates a subset of retain data to mitigate performance degradation. This group includes descent-based variants of the aforementioned methods: GradAsc Descent, AGE Descent, and UNDIAL Descent. In addition, we include GradRest Descent (Ko et al., 2024).

Large Language Model (LLM) Baselines. For the concept unlearning task on Large Language Models, we compare our CURE framework against a suite of relevant methods. The baselines include GradAsc Descent, UNDIAL (Dong et al., 2024), GradDiff Yao et al. (2024), Negative Preference Optimization (NPO) Zhang et al. (2024), and its recent variant SimNPO Fan et al. (2025). To ensure a fair and direct comparison, all baseline methods were implemented using a retain set.

A.3 Implementation details and hyperparameters.

All unlearning procedures were implemented using a Stochastic Gradient Descent (SGD) optimizer, typically by minimizing a loss function or, in the case of Gradient Ascent, by minimizing a negative loss to perform ascent. To ensure a fair comparison of retain accuracy, the learning rate for each method was tuned within the range of $1\text{e}{-4}$ to $1\text{e}{-5}$ to achieve the lowest possible forget accuracy, thereby establishing the strongest possible performance for each unlearning model. For most experimental settings, including the retain-data-aided case and baselines in the forget-data-only case, we trained for 10 epochs, as performance was observed to converge around the 6th epoch. To ensure full convergence for our GROGU method in the forget-data-only setting, we extended its training to 30 epochs on CIFAR-10/100 and 20 on ImageNet. This extended training remains highly efficient due to operating only on the small forget set and is justified by the superior results it enables while other unlearn methods cannot. Finally, for GSR, we found that the optimal distillation temperature T increased significantly with dataset complexity, rising from $T = 12$ on CIFAR-10 to $T = 70$ on ImageNet. For GROGU, as dataset complexity increased, the optimal weight for the soft-label objective systematically shifted from 0.5 on CIFAR-10 to 0.99 on ImageNet. This trend underscores the growing importance of soft-label distillation for preserving knowledge in complex data hierarchies. For our LLM experiments, we evaluated three distinct hyperparameter configurations for our GROGU method. The first was a low-temperature setting with $T = 1$ and equal hard and soft objective weights of 0.5. The second used a higher temperature of $T = 5$ with the same 0.5 weights. The final configuration was a high-weight ablation, also at $T = 5$, where both objective weights were increased to 10.

A.4 CURE ALGORITHMS

Algorithm 1 Guided Hard Relabel (GHR)

```
Require: Forget set $\mathcal{D}_f$, original (frozen) model $f_{\theta_o}$, trainable model $f_\theta$, class to forget $u$, optimizer Opt, batch size $B$
Define cross-entropy $\ell_{\mathrm{CE}}(f_\theta(x), y) = -\log f_\theta(x)_y$
for each training step do
    Sample mini-batch $\{x_j\}_{j=1}^{B} \subset \mathcal{D}_f$
    for $j = 1$ to $B$ do
        $r_j \leftarrow \arg\max_{k \neq u} \big(f_{\theta_o}(x_j)\big)_k$          ▷ guided hard label
    end for
    $\mathcal{L}_{\mathrm{GHR}} \leftarrow \frac{1}{B}\sum_{j=1}^{B} \ell_{\mathrm{CE}}\big(f_\theta(x_j),\, r_j\big)$
    $\theta \leftarrow \mathrm{Opt}\big(\theta,\, \nabla_\theta \mathcal{L}_{\mathrm{GHR}}\big)$
end for
```

Algorithm 2 Guided Soft Relabel (GSR)

```
Require: Forget set $\mathcal{D}_f$, original (frozen) model $f_{\theta_o}$, trainable model $f_\theta$, class to forget $u$, temperature $T > 0$, masking constant $C \ll 0$ (e.g., $-10^9$), optimizer Opt, batch size $B$
Teacher targets: $\tilde{z}(x) \leftarrow f_{\theta_o}(x)$;   $\tilde{z}_u(x) \leftarrow C$;   $\mathbf{p}_T(x) \leftarrow \mathrm{softmax}(\tilde{z}(x)/T)$
Student probs: $\mathbf{q}_T(x;\theta) \leftarrow \mathrm{softmax}(f_\theta(x)/T)$
Loss: $\ell_{\mathrm{KD}}(x) \leftarrow T^2\, D_{\mathrm{KL}}\big(\mathbf{p}_T(x) \,\|\, \mathbf{q}_T(x;\theta)\big)$
for each training step do
    Sample mini-batch $\{x_j\}_{j=1}^{B} \subset \mathcal{D}_f$
    $\mathcal{L}_{\mathrm{GSR}} \leftarrow \frac{1}{B}\sum_{j=1}^{B} \ell_{\mathrm{KD}}(x_j)$
    $\theta \leftarrow \mathrm{Opt}\big(\theta,\, \nabla_\theta \mathcal{L}_{\mathrm{GSR}}\big)$
end for
```

Algorithm 3 Guided Restricted Orthogonal Gradient Unlearning (GROGU)

```
Require: Forget set $\mathcal{D}_f$, $f_{\theta_o}$ (frozen), $f_\theta$ (trainable), class $u$, temperature $T$, constant $C \ll 0$, optimizer Opt, batch size $B$, small $\varepsilon > 0$
Define $\mathrm{Proj}_b^{\perp}(a) := a - \frac{\langle a,b\rangle}{\|b\|_2^2+\varepsilon}\, b$
for each training step do
    Sample mini-batch $\{x_j\}_{j=1}^{B} \subset \mathcal{D}_f$
    for $j = 1$ to $B$ do
        GHR label: $r_j \leftarrow \arg\max_{k \neq u} \big(f_{\theta_o}(x_j)\big)_k$
        GSR target: $\tilde{z}(x_j) \leftarrow f_{\theta_o}(x_j)$;   $\tilde{z}_u(x_j) \leftarrow C$;   $\mathbf{p}_T(x_j) \leftarrow \mathrm{softmax}(\tilde{z}(x_j)/T)$; $\mathbf{q}_T(x_j;\theta) \leftarrow \mathrm{softmax}(f_\theta(x_j)/T)$
        Per-sample losses: $\ell_{\mathrm{H},j} \leftarrow \ell_{\mathrm{CE}}\big(f_\theta(x_j), r_j\big)$,   $\ell_{\mathrm{S},j} \leftarrow T^2 D_{\mathrm{KL}}\big(\mathbf{p}_T(x_j) \,\|\, \mathbf{q}_T(x_j;\theta)\big)$
        Gradients: $g_{H,j} \leftarrow \nabla_\theta \ell_{\mathrm{H},j}$,   $g_{S,j} \leftarrow \nabla_\theta \ell_{\mathrm{S},j}$
        Restricted orthogonal combination: $\hat{g}_{H,j} \leftarrow \mathrm{Proj}^{\perp}_{g_{S,j}}(g_{H,j})$,   $\hat{g}_{S,j} \leftarrow \mathrm{Proj}^{\perp}_{g_{H,j}}(g_{S,j})$
        $g_{\mathrm{GROGU},j} \leftarrow \hat{g}_{H,j} + \hat{g}_{S,j}$
    end for
    $\mathcal{G} \leftarrow \frac{1}{B}\sum_{j=1}^{B} g_{\mathrm{GROGU},j}$          ▷ batch-averaged update
    $\theta \leftarrow \mathrm{Opt}\big(\theta,\, \mathcal{G}\big)$
end for
```

A.5 Full comparison results with baseline methods on CIFAR-10

This section presents the comprehensive experimental results for all unlearning methods on the CIFAR-10 dataset.

First, we analyze the overall performance metrics, detailed across three different forget-class scenarios. As summarized in the main results table 3, a key initial finding on CIFAR-10 is that most methods successfully achieve near-zero forget accuracy. This is likely attributable to the dataset's relatively simple 10-class structure and the sufficient size of the forget set. In the more challenging scenario using forget data only, our proposed GROGU and GROGU T1 methods consistently outperform all baselines, demonstrating the highest retain accuracy while successfully erasing the target class. When retain data is incorporated (Descent methods), our CURE framework methods (GHR Descent, GSR Descent, and GROGU Descent) collectively outperform competing methods like GradAsc Descent, GradRest Descent, AGE Descent, and UNDIAL Descent. Among our CURE methods, GHR Descent proves to be particularly robust, achieving the top performance in this setting across all forget scenarios.

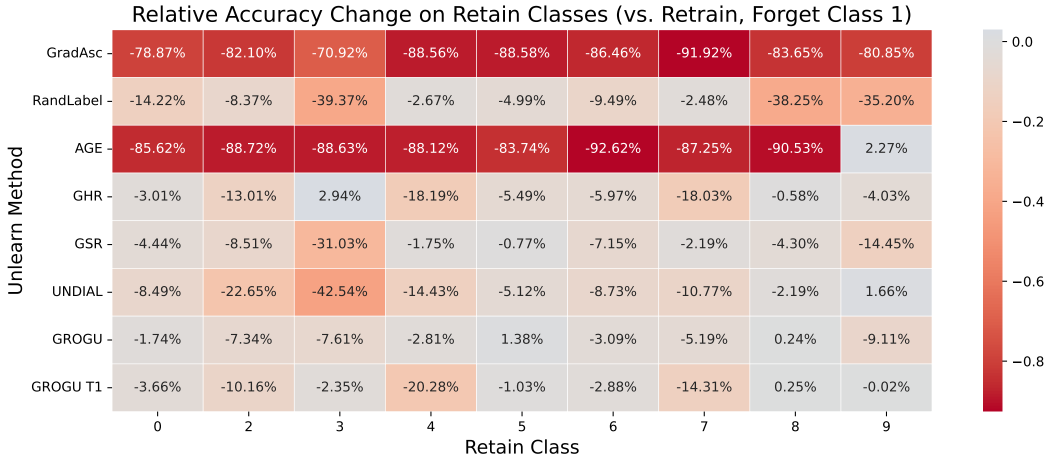
Second, a more in-depth analysis of the impact on retained classes is provided by the accuracy drop heatmaps figures 3-8. Note that Finetune is excluded from this analysis due to its unacceptably high forget accuracy, as shown in the main results table. Across all scenarios, the heatmaps visually confirm that our CURE methods result in lighter-colored rows, signifying less collateral damage to retain-set performance. Specifically, in the forget-data-only setting, GROGU preserves accuracy on the most relevant retain class (e.g., class 9 when forgetting class 1 as in figure 3) as well as general retain classes accuracy. This observation holds true across the different forget-class experiments.

Third, the prediction count distributions figures 9-14 reveal how well each method emulates the behavior of the Retrain model. When given images from the forget class, our CURE methods (GHR, GSR, GROGU), along with baselines AGE and UNDIAL, successfully redirect predictions towards semantically related classes, closely mirroring the Retrain model. This is in stark contrast to methods like GradAsc and RandLabel, which scatter predictions more broadly. We also observe the effect of the distillation temperature: increasing temperature in GSR and GROGU causes a shift from focusing on one relevant class to preserving all retain classes more generally. Notably, GROGU maintains a stronger alignment with the Retrain model's prediction pattern even at higher temperatures compared to GSR.

Finally, we evaluate how closely the unlearning updates align with a full retraining by measuring the Euclidean distance between input-pixel gradients in figures 15-20. The distributions show that models produced by our CURE framework, as well as by AGE and UNDIAL, are significantly closer to the Retrain model in gradient space than naive methods like GradAsc and RandLabel are. This indicates that our framework successfully captures the beneficial optimization characteristics of these advanced baselines, while simultaneously achieving the superior retain-set accuracy established in our primary results table.

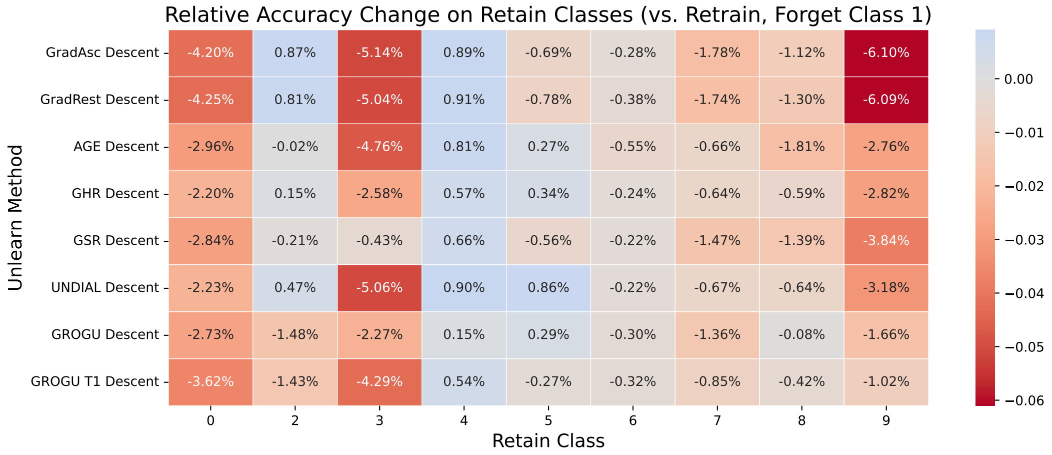
Table 3: This table presents a full comparison of unlearning methods on CIFAR-10, where all metrics are the mean performance over 30 random seeds. We report performance across three distinct forget-class experiments. For each experiment, we show the overall test accuracy (Overall Acc.), the accuracy on the 9 retained classes (Retain Acc.), and the accuracy on the single forgotten class (Forget Acc.). The best-performing method in each subgroup is highlighted in bold, selected first by achieving a Forget Acc. $\leq$ 0.003, and then by having the highest Retain Acc.

Method	Overall Acc.	Retain Acc.	Forget Acc.
Forgetting Class 1			
Reference Models			
Original	0.930 ± 0.003	0.926 ± 0.003	0.961 ± 0.006
Retrain	0.833 ± 0.002	0.929 ± 0.003	0.000 ± 0.000
— Unlearning with Forget Data Only —			
GradAsc	0.137 ± 0.018	0.152 ± 0.020	0.000 ± 0.000
RandLabel	0.693 ± 0.078	0.768 ± 0.087	0.044 ± 0.033
Finetune	0.845 ± 0.014	0.928 ± 0.002	0.119 ± 0.133
AGE	0.194 ± 0.194	0.209 ± 0.208	0.061 ± 0.094
GHR	0.773 ± 0.018	0.862 ± 0.020	0.000 ± 0.001
GSR	0.766 ± 0.018	0.854 ± 0.020	0.000 ± 0.001
UNDIAL	0.733 ± 0.040	0.817 ± 0.045	0.000 ± 0.000
GROGU	**0.801 ± 0.006**	**0.893 ± 0.007**	**0.000 ± 0.001**
GROGU T1	0.784 ± 0.034	0.874 ± 0.038	0.002 ± 0.014
— Unlearning with Retain Data —			
GradAsc Descent	0.817 ± 0.003	0.910 ± 0.004	0.001 ± 0.009
GradRest Descent	0.817 ± 0.004	0.910 ± 0.004	0.001 ± 0.011
AGE Descent	0.822 ± 0.003	0.916 ± 0.004	0.000 ± 0.000
GHR Descent	**0.826 ± 0.003**	**0.920 ± 0.003**	**0.000 ± 0.000**
GSR Descent	0.823 ± 0.003	0.918 ± 0.003	0.000 ± 0.000
UNDIAL Descent	0.824 ± 0.003	0.919 ± 0.003	0.000 ± 0.000
GROGU Descent	0.825 ± 0.003	0.919 ± 0.003	0.000 ± 0.000
GROGU T1 Descent	0.823 ± 0.003	0.917 ± 0.004	0.000 ± 0.000
Forgetting Class 3			
Reference Models			
Original	0.930 ± 0.003	0.926 ± 0.003	0.961 ± 0.006
Retrain	0.854 ± 0.002	0.949 ± 0.002	0.000 ± 0.000
— Unlearning with Forget Data Only —			
GradAsc	0.132 ± 0.017	0.146 ± 0.019	0.000 ± 0.000
RandLabel	0.589 ± 0.076	0.645 ± 0.084	0.082 ± 0.020
Finetune	0.856 ± 0.003	0.949 ± 0.002	0.023 ± 0.022
AGE	0.124 ± 0.034	0.123 ± 0.033	0.132 ± 0.120
GHR	0.831 ± 0.013	0.923 ± 0.015	0.000 ± 0.001
GSR	0.798 ± 0.036	0.886 ± 0.041	0.001 ± 0.002
UNDIAL	0.837 ± 0.010	0.930 ± 0.012	0.000 ± 0.000
GROGU	0.836 ± 0.014	0.929 ± 0.015	0.000 ± 0.000
GROGU T1	**0.846 ± 0.003**	**0.940 ± 0.004**	**0.000 ± 0.000**
— Unlearning with Retain Data —			
GradAsc Descent	0.845 ± 0.004	0.934 ± 0.004	0.052 ± 0.020
GradRest Descent	0.846 ± 0.004	0.933 ± 0.003	0.056 ± 0.018
AGE Descent	0.850 ± 0.003	0.944 ± 0.003	0.001 ± 0.002
GHR Descent	**0.852 ± 0.002**	**0.946 ± 0.002**	**0.000 ± 0.000**
GSR Descent	0.851 ± 0.002	0.945 ± 0.002	0.000 ± 0.001
UNDIAL Descent	0.847 ± 0.002	0.941 ± 0.003	0.000 ± 0.001
GROGU Descent	**0.852 ± 0.002**	**0.946 ± 0.003**	**0.000 ± 0.001**
GROGU T1 Descent	**0.851 ± 0.002**	**0.946 ± 0.003**	**0.000 ± 0.000**
Forgetting Class 8			
Reference Models			
Original	0.930 ± 0.003	0.926 ± 0.003	0.961 ± 0.006
Retrain	0.834 ± 0.002	0.929 ± 0.002	0.000 ± 0.000
— Unlearning with Forget Data Only —			
GradAsc	0.134 ± 0.016	0.149 ± 0.018	0.000 ± 0.000
RandLabel	0.599 ± 0.097	0.663 ± 0.108	0.038 ± 0.023
Finetune	0.850 ± 0.014	0.927 ± 0.002	0.174 ± 0.139
AGE	0.121 ± 0.018	0.132 ± 0.019	0.021 ± 0.047
GHR	0.769 ± 0.016	0.857 ± 0.018	0.000 ± 0.000
GSR	0.552 ± 0.103	0.614 ± 0.114	0.000 ± 0.001
UNDIAL	0.576 ± 0.063	0.641 ± 0.070	0.002 ± 0.003
GROGU	0.764 ± 0.021	0.851 ± 0.023	0.000 ± 0.000
GROGU T1	**0.784 ± 0.016**	**0.873 ± 0.018**	**0.000 ± 0.000**
— Unlearning with Retain Data —			
GradAsc Descent	0.815 ± 0.005	0.901 ± 0.006	0.063 ± 0.045
GradRest Descent	0.817 ± 0.005	0.902 ± 0.004	0.070 ± 0.047
AGE Descent	0.819 ± 0.005	0.912 ± 0.005	0.000 ± 0.001
GHR Descent	**0.823 ± 0.003**	**0.917 ± 0.003**	**0.000 ± 0.000**
GSR Descent	0.818 ± 0.003	0.911 ± 0.004	0.000 ± 0.000
UNDIAL Descent	0.819 ± 0.003	0.912 ± 0.004	0.000 ± 0.000
GROGU Descent	0.818 ± 0.003	0.911 ± 0.003	0.000 ± 0.000
GROGU T1 Descent	0.818 ± 0.004	0.912 ± 0.004	0.000 ± 0.000



Unlearn Method	0	2	3	4	5	6	7	8	9
GradAsc	-78.87%	-82.10%	-70.92%	-88.56%	-88.58%	-86.46%	-91.92%	-83.65%	-80.85%
RandLabel	-14.22%	-8.37%	-39.37%	-2.67%	-4.99%	-9.49%	-2.48%	-38.25%	-35.20%
AGE	-85.62%	-88.72%	-88.63%	-88.12%	-83.74%	-92.62%	-87.25%	-90.53%	2.27%
GHR	-3.01%	-13.01%	2.94%	-18.19%	-5.49%	-5.97%	-18.03%	-0.58%	-4.03%
GSR	-4.44%	-8.51%	-31.03%	-1.75%	-0.77%	-7.15%	-2.19%	-4.30%	-14.45%
UNDIAL	-8.49%	-22.65%	-42.54%	-14.43%	-5.12%	-8.73%	-10.77%	-2.19%	1.66%
GROGU	-1.74%	-7.34%	-7.61%	-2.81%	1.38%	-3.09%	-5.19%	0.24%	-9.11%
GROGU T1	-3.66%	-10.16%	-2.35%	-20.28%	-1.03%	-2.88%	-14.31%	0.25%	-0.02%

Figure 3: Heatmap of per-class accuracy drop on the CIFAR-10 retain set after unlearning class 1. Each row represents an unlearning method with forget data only, and each column corresponds to one of the 9 retain classes. The color of each cell visualizes the change in accuracy compared to the fully retrained model. Deep red indicates a large drop in accuracy, while gray and blue indicate that performance is preserved or even improved. Lighter colors across a row demonstrate superior performance preservation.



Unlearn Method	0	2	3	4	5	6	7	8	9
GradAsc Descent	-4.20%	0.87%	-5.14%	0.89%	-0.69%	-0.28%	-1.78%	-1.12%	-6.10%
GradRest Descent	-4.25%	0.81%	-5.04%	0.91%	-0.78%	-0.38%	-1.74%	-1.30%	-6.09%
AGE Descent	-2.96%	-0.02%	-4.76%	0.81%	0.27%	-0.55%	-0.66%	-1.81%	-2.76%
GHR Descent	-2.20%	0.15%	-2.58%	0.57%	0.34%	-0.24%	-0.64%	-0.59%	-2.82%
GSR Descent	-2.84%	-0.21%	-0.43%	0.66%	-0.56%	-0.22%	-1.47%	-1.39%	-3.84%
UNDIAL Descent	-2.23%	0.47%	-5.06%	0.90%	0.86%	-0.22%	-0.67%	-0.64%	-3.18%
GROGU Descent	-2.73%	-1.48%	-2.27%	0.15%	0.29%	-0.30%	-1.36%	-0.08%	-1.66%
GROGU T1 Descent	-3.62%	-1.43%	-4.29%	0.54%	-0.27%	-0.32%	-0.85%	-0.42%	-1.02%

Figure 4: Heatmap of per-class accuracy drop on the CIFAR-10 retain set after unlearning class 1. Each row represents an unlearning method with forget and retain data, and each column corresponds to one of the 9 retain classes.

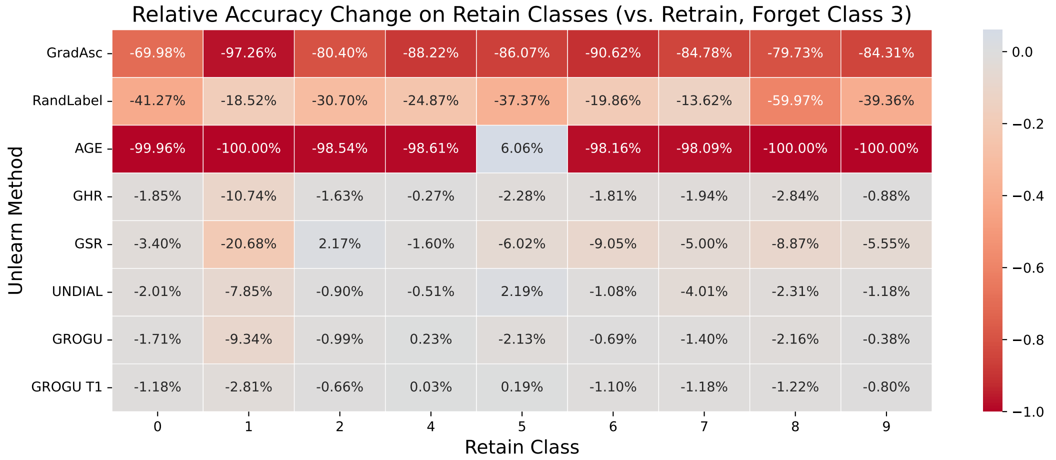


Figure 5: Heatmap of per-class accuracy drop on the CIFAR-10 retain set after unlearning class 3. Each row represents an unlearning method with forget data only, and each column corresponds to one of the 9 retain classes.

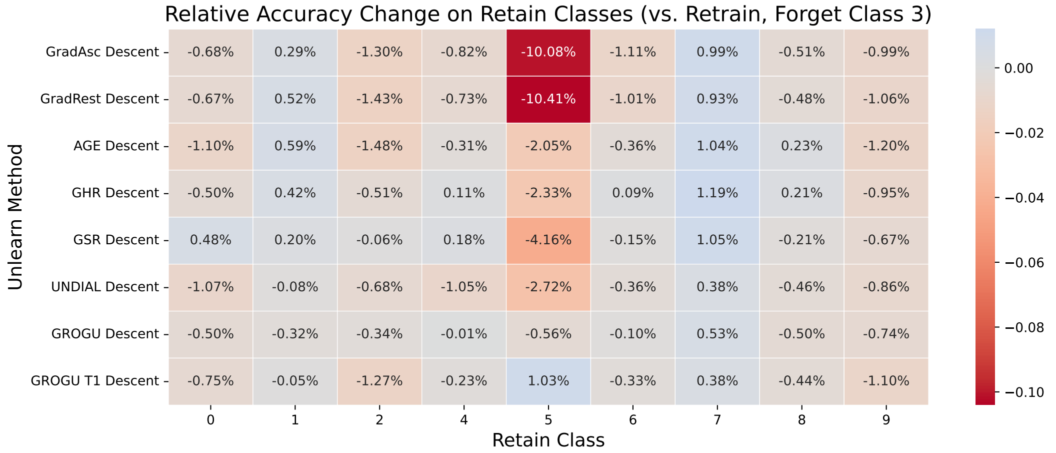


Figure 6: Heatmap of per-class accuracy drop on the CIFAR-10 retain set after unlearning class 3. Each row represents an unlearning method with forget and retain data, and each column corresponds to one of the 9 retain classes.

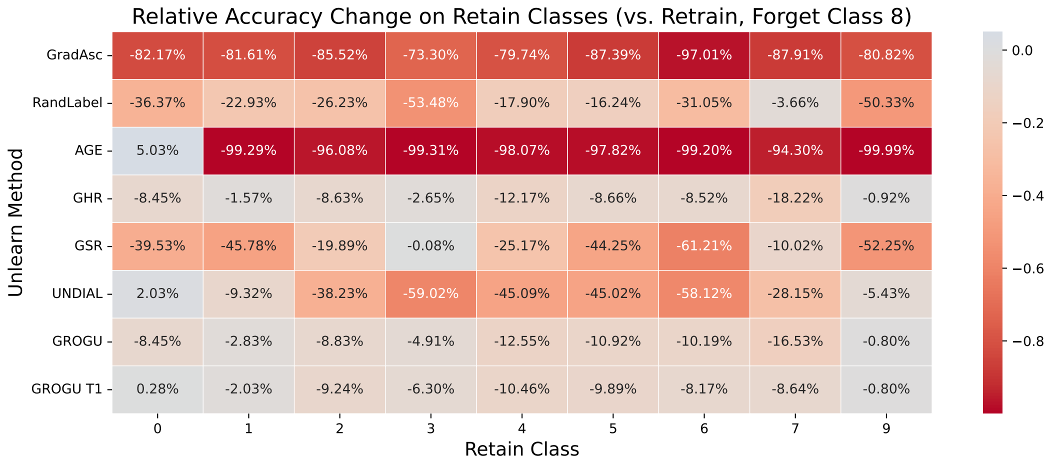


Figure 7: Heatmap of per-class accuracy drop on the CIFAR-10 retain set after unlearning class 8. Each row represents an unlearning method with forget data only, and each column corresponds to one of the 9 retain classes.

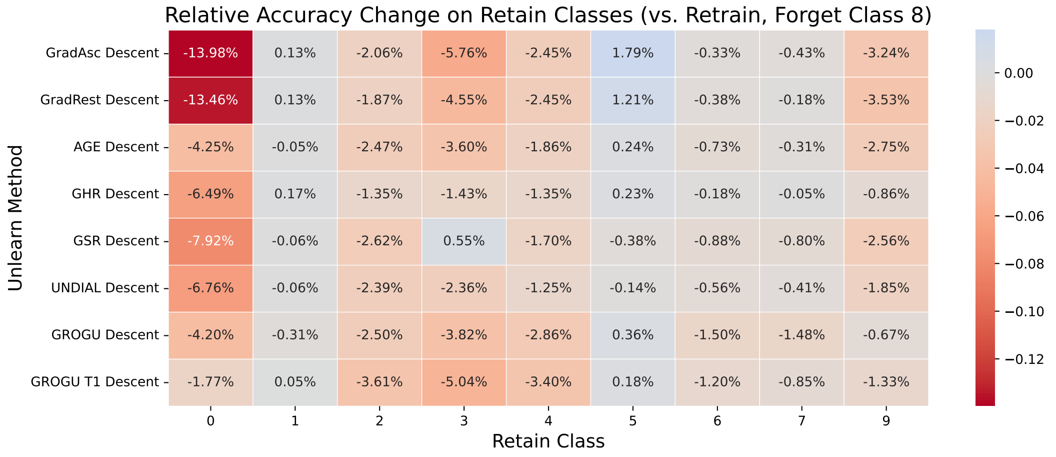


Figure 8: Heatmap of per-class accuracy drop on the CIFAR-10 retain set after unlearning class 8. Each row represents an unlearning method with forget and retain data, and each column corresponds to one of the 9 retain classes.

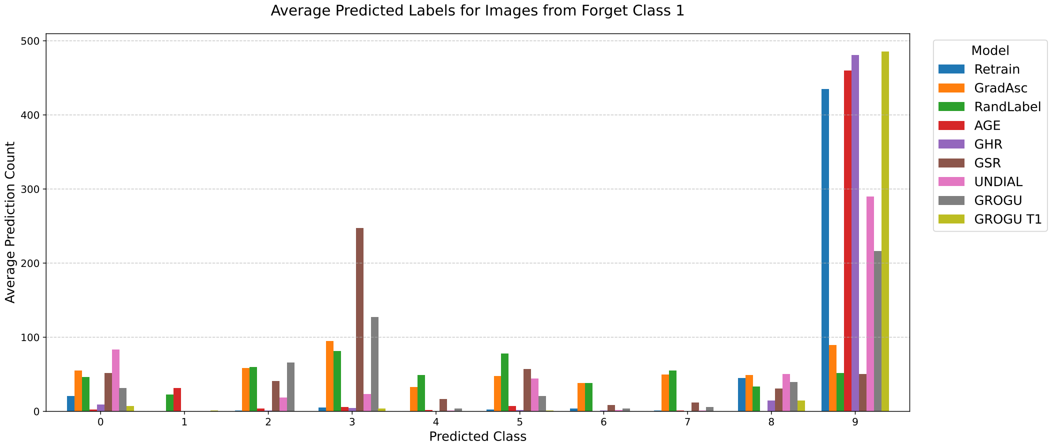


Figure 9: Prediction counts for images from the forgotten class 1 when processed by different unlearned models with forget data only.

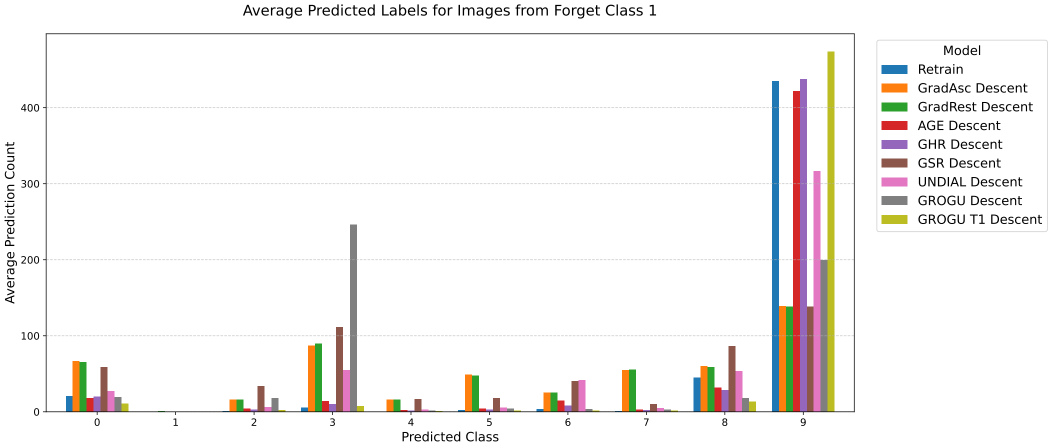


Figure 10: Prediction counts for images from the forgotten class 1 when processed by different unlearned models incorporate retain loss in addition to forget loss.

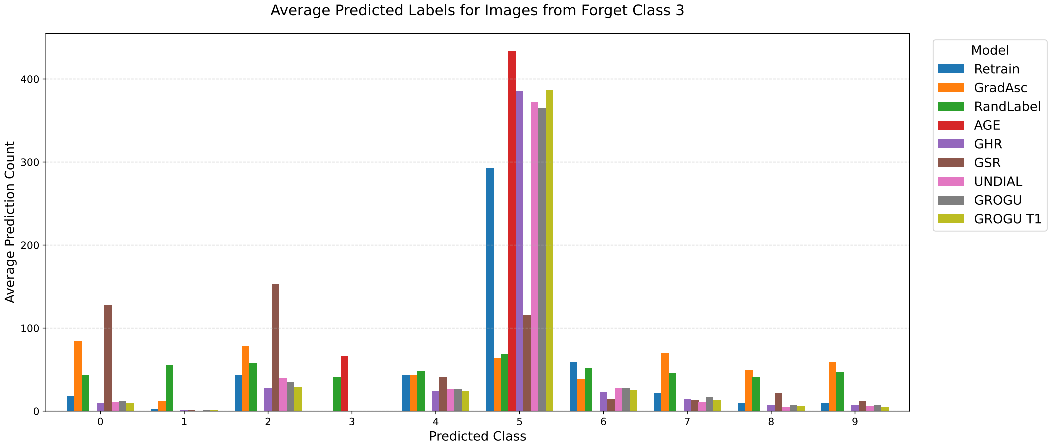


Figure 11: Prediction counts for images from the forgotten class 3 when processed by different unlearned models with forget data only.

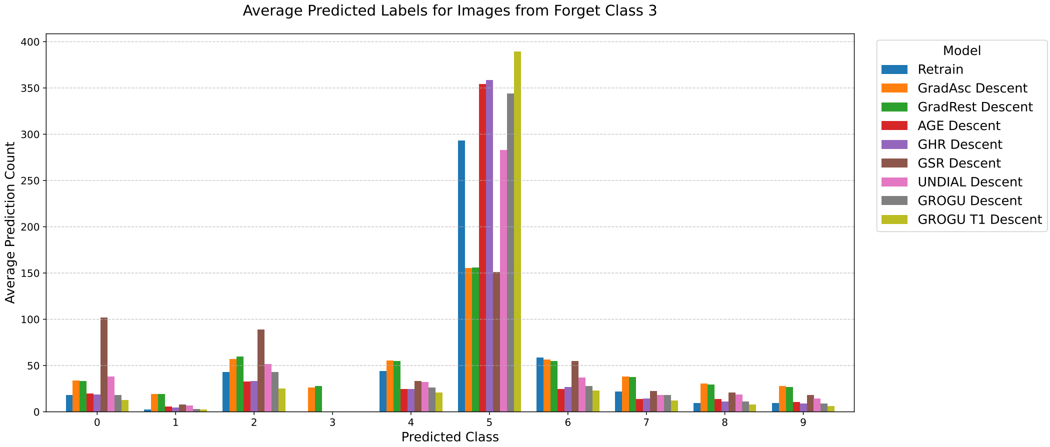


Figure 12: Prediction counts for images from the forgotten class 3 when processed by different unlearned models incorporate retain loss in addition to forget loss.

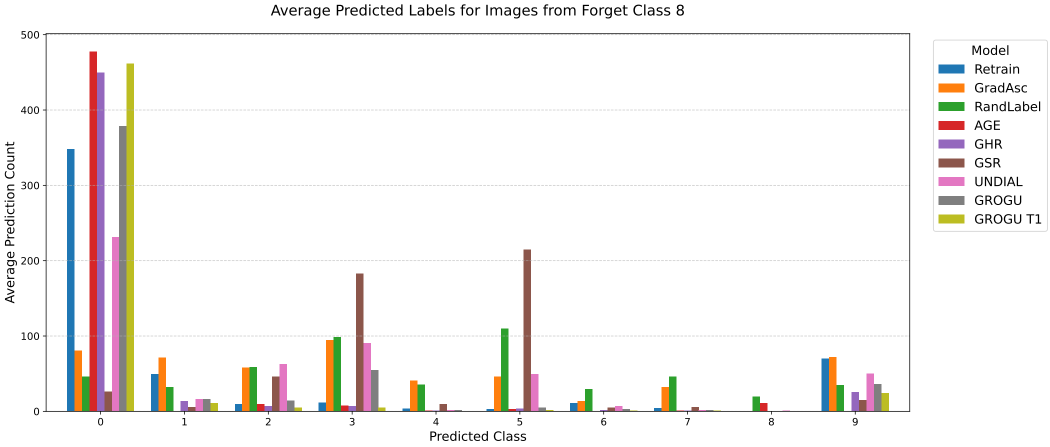


Figure 13: Prediction counts for images from the forgotten class 8 when processed by different unlearned models with forget data only.

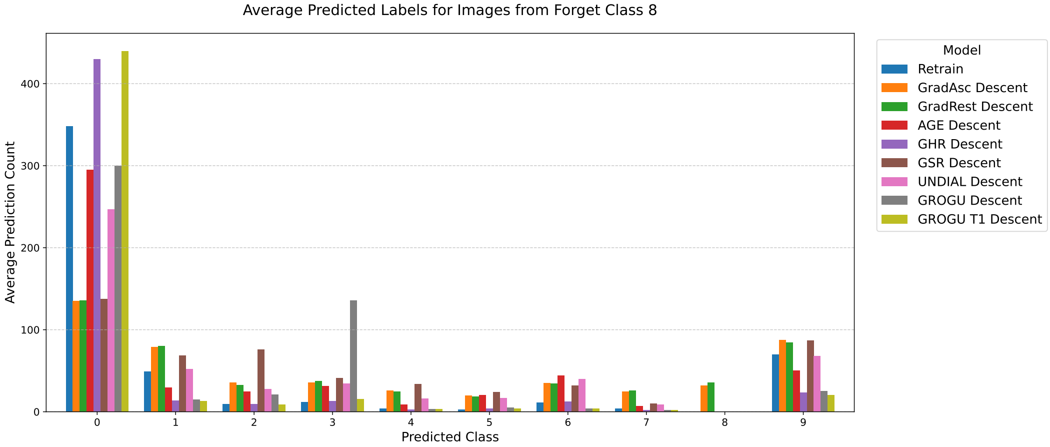


Figure 14: Prediction counts for images from the forgotten class 8 when processed by different unlearned models incorporate retain loss in addition to forget loss.

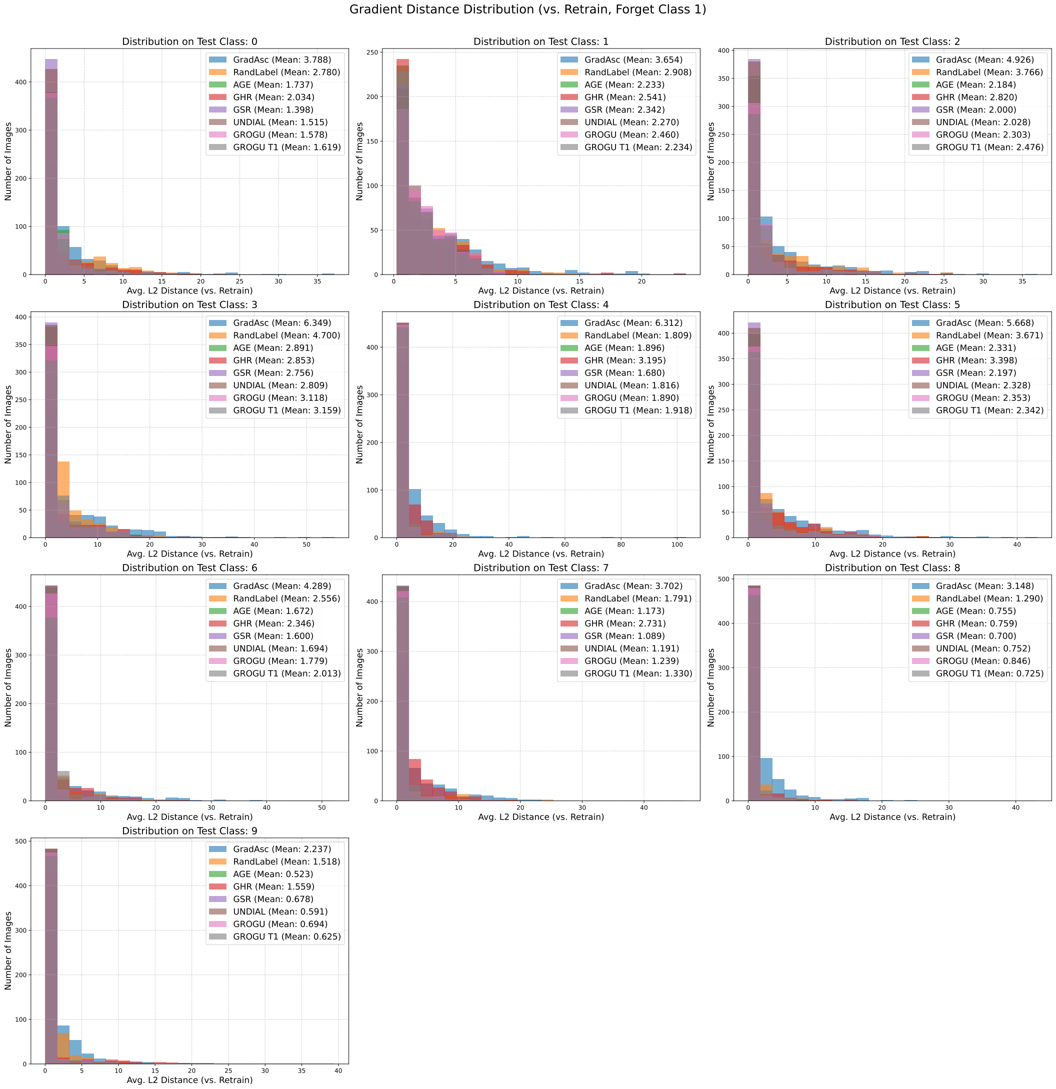


Figure 15: Euclidean distance between the input gradients of unlearn models which is trained with forget class 1 data only and the retrained model.

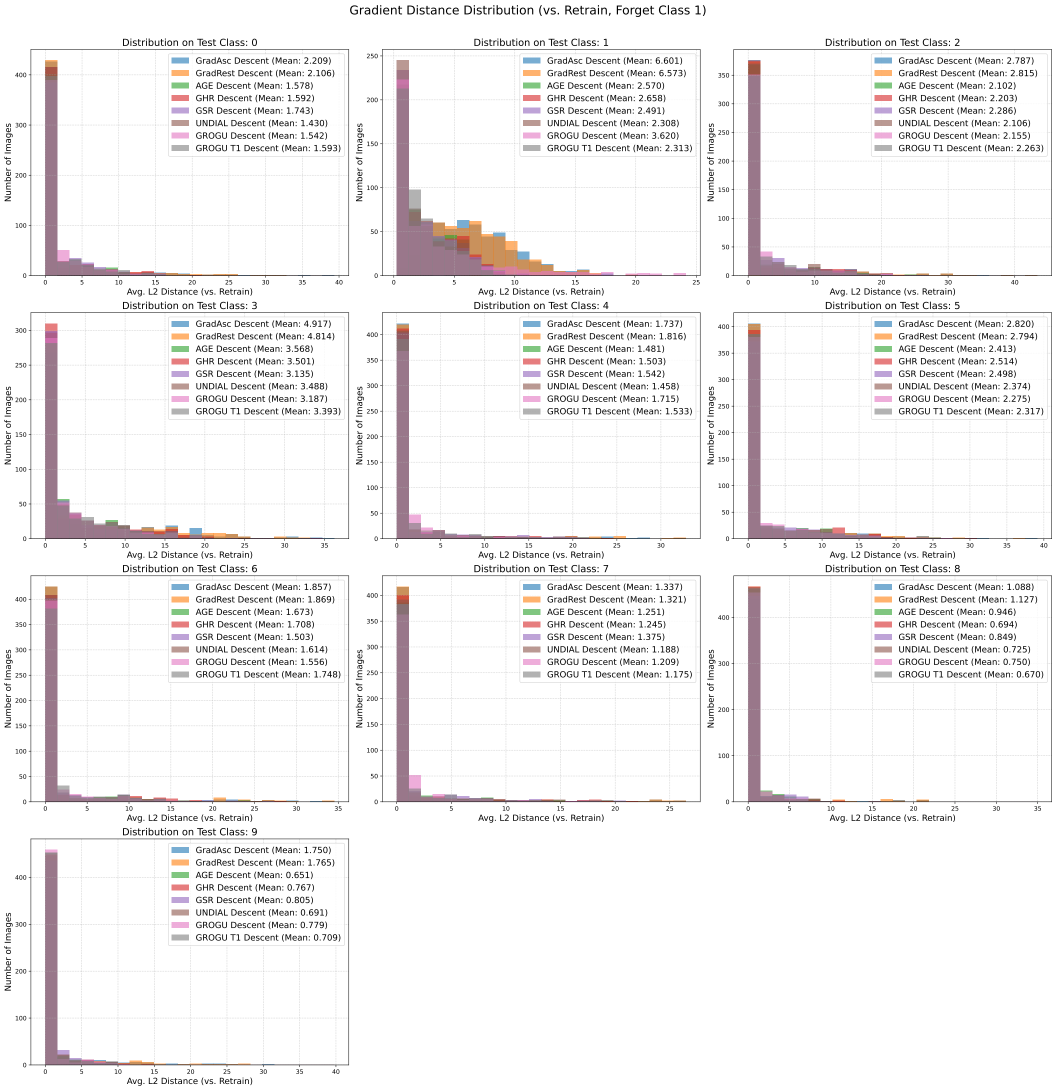


Figure 16: Euclidean distance between the input gradients of unlearn models and the retrained model.

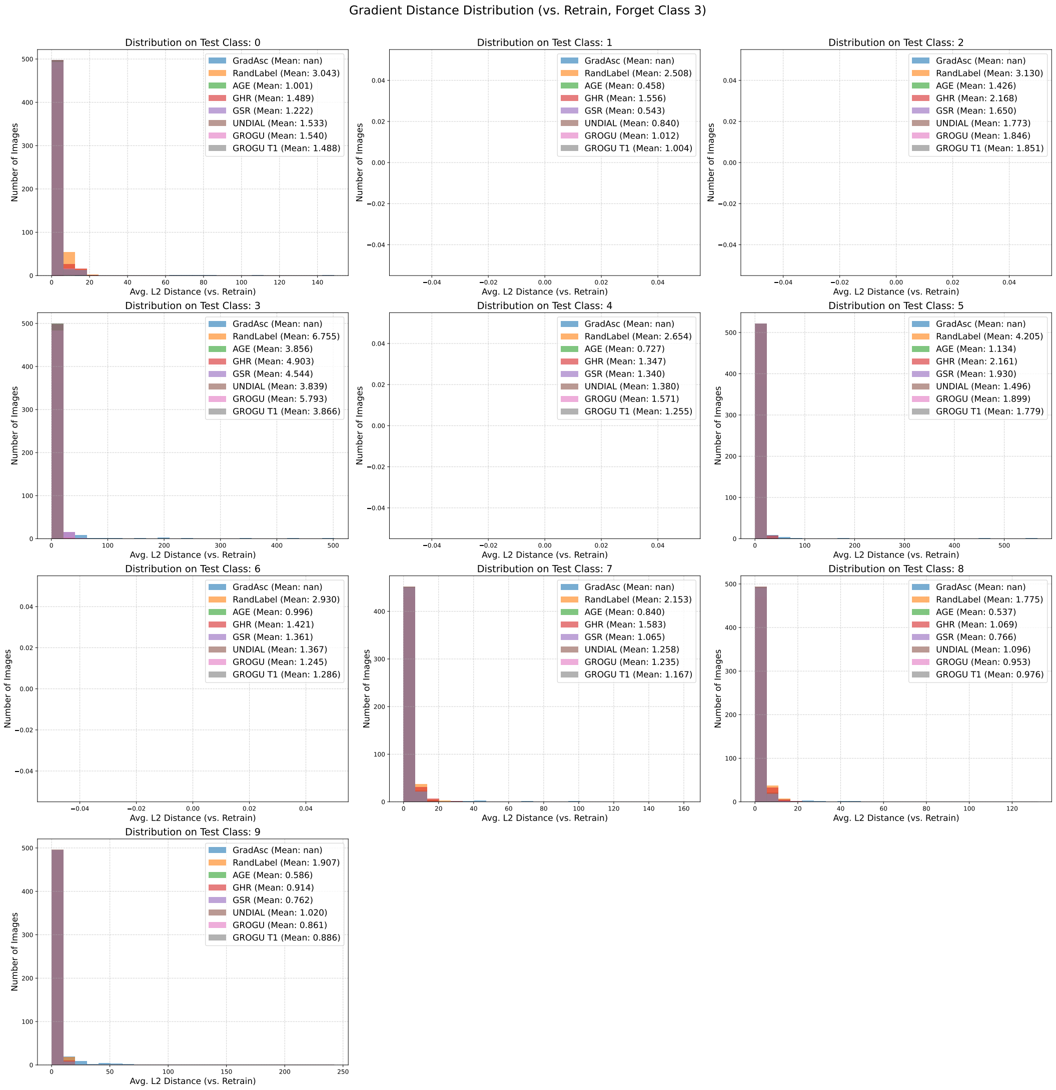


Figure 17: Euclidean distance between the input gradients of unlearn models which is trained with forget class 3 data only and the retrained model. The subplot for GradAsc is empty because the method produced NaN values during gradient generation. This is a direct result of its unstable optimization process.

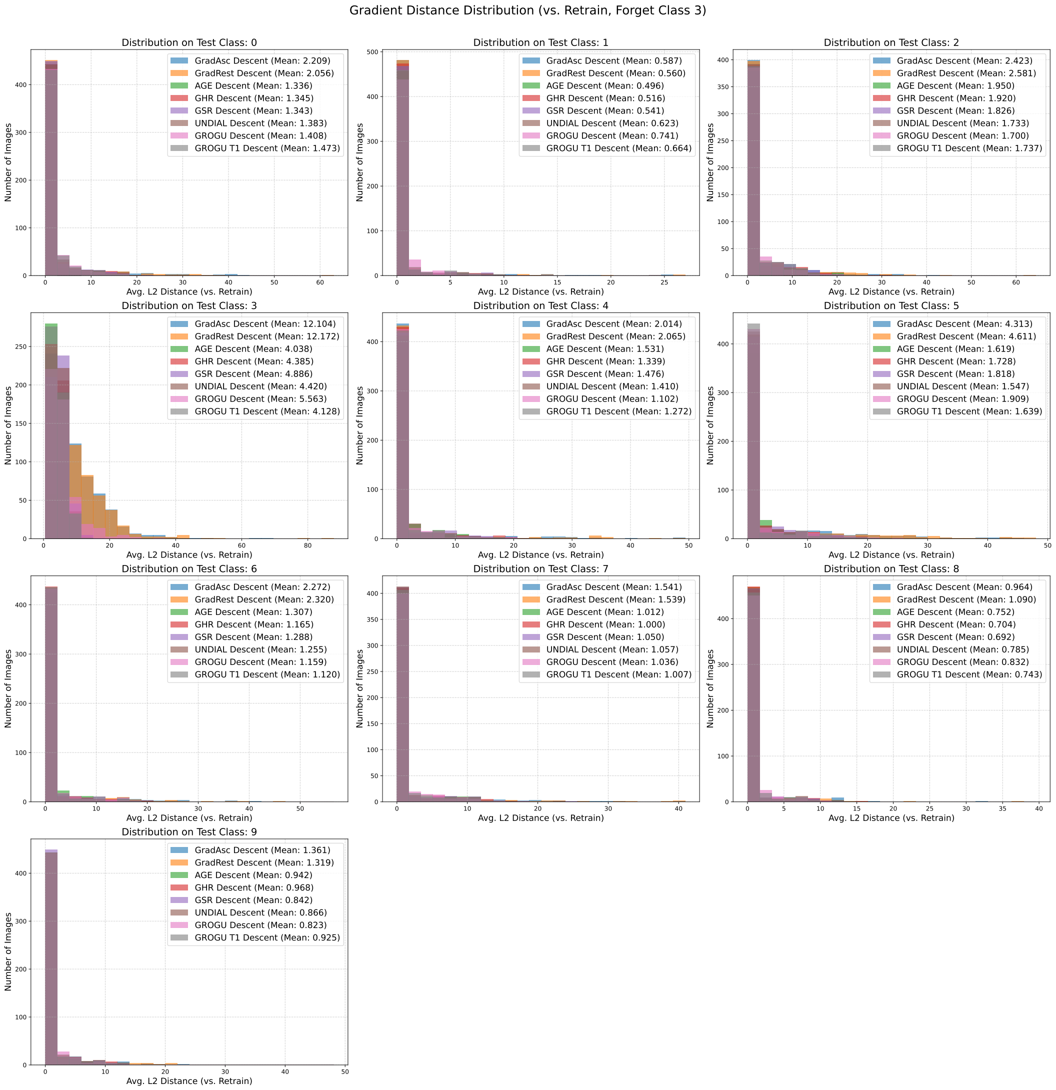


Figure 18: Euclidean distance between the input gradients of unlearn models and the retrained model.

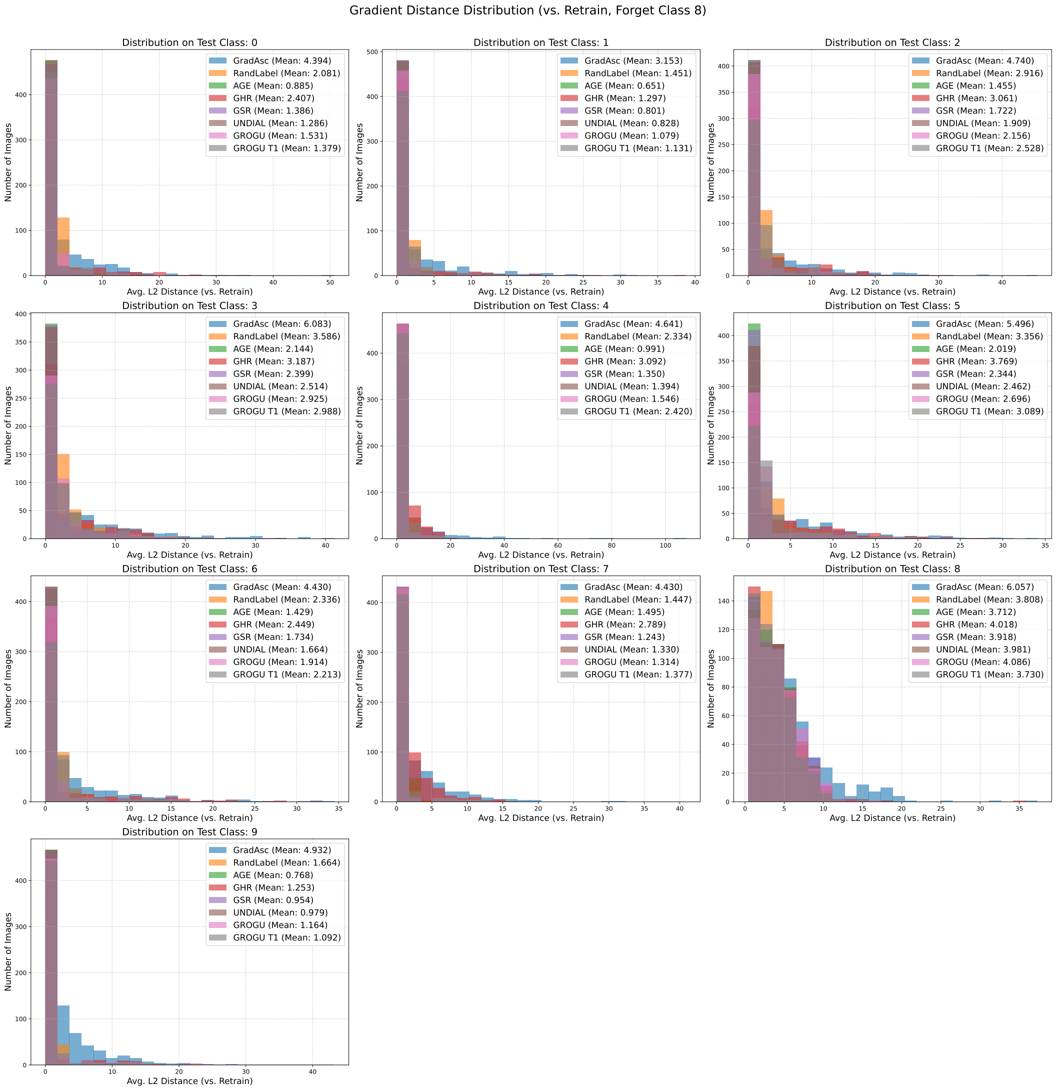


Figure 19: Euclidean distance between the input gradients of unlearn models which is trained with forget class 8 data only and the retrained model.

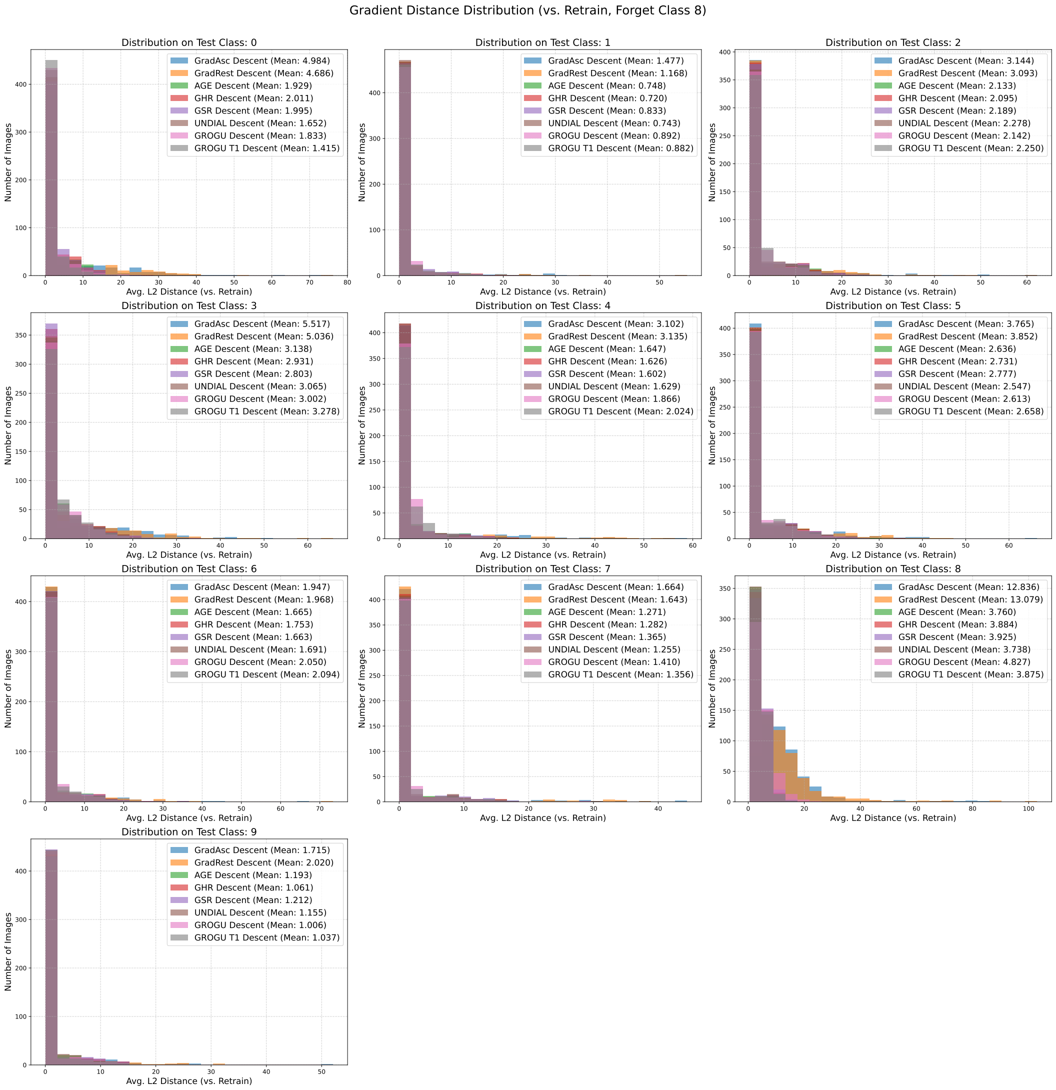


Figure 20: Euclidean distance between the input gradients of unlearn models and the retrained model.

A.6 Full comparison results with baseline methods on CIFAR-100

This section details the results on the more challenging CIFAR-100 dataset. As we observe across all forget classes in table 4, achieving a perfect zero forget accuracy is more difficult on CIFAR-100. This is expected, as the dataset contains the same number of images as CIFAR-10 but spread across 100 classes, resulting in a much smaller set of forget-class images available for unlearning.

The main results table 4 shows that incorporating retain data generally leads to lower forget accuracy. In the unlearning scenario with only forget data, our proposed GROGU method consistently stands out across all forget classes, achieving the lowest forget accuracy while simultaneously maintaining the highest retain accuracy among all competitors.

When a limited subset of retain data is used, the analysis becomes more nuanced. In the experiments for forgetting classes 16 and 54, if we consider a practical forget accuracy threshold of less than 0.010, our CURE method GSR Descent achieves the highest retain accuracy. It outperforms competing methods like AGE Descent and UNDIAL Descent, which not only have lower retain accuracy but also a higher forget accuracy. In the experiment for forgetting class 81, while GSR Descent and UNDIAL Descent show the same average retain accuracy, our GSR Descent is superior due to its lower forget accuracy.

A more in-depth analysis of per-class performance is provided by the accuracy drop heatmaps in figures 21-32, where Finetune is excluded due to its high forget accuracy. In the forget-data-only setting, the general heatmap figures 22, 26, and 30 reveals that `GROGU` and `GROGU T1` exhibit a more uniform performance profile, avoiding the severe, isolated accuracy drops (dark red cells) seen in other methods like GHR and UNDIAL. When we zoom in on a focused subset of semantically relevant retain classes (defined as classes to which the original model misclassifies more than two forget-class images on average over 30 seeds of models), in figures 21, 23, and 25, we find that GROGU leads a much smaller accuracy drop than GSR. In summary, these heatmaps reveal a key trade-off: GSR preserves general classes at the cost of relevant ones, while GHR and UNDIAL preserve relevant classes at the cost of general ones. GROGU successfully resolves this trade-off, preserving both relevant and general classes more effectively than any other method in this challenging setting in which retain data is not available. When retain data is incorporated, we find that the CURE method GSR Descent alone is sufficient to achieve strong performance across both relevant and general classes.

Finally, we analyze the prediction count distributions to see how well each method emulates the Retrain model in figures 34-38. We observe a clear hierarchy of alignment. AGE and GHR most closely match the prediction pattern of the retrained model, both with and without retain data. Our GROGU and GROGU T1 methods follow, also showing strong alignment. Methods like GSR and UNDIAL show a lesser degree of alignment, while GradAsc, RandLabel, GradAsc Descent, GradRest Descent exhibit the least alignment with the retrained model's behavior.

Table 4: This table presents a full comparison of unlearning methods on CIFAR-100, where all metrics are the mean performance over 30 random seeds. We report performance across three distinct forget-class experiments. For each experiment, we show the overall test accuracy (Overall Acc.), the accuracy on the 9 retained classes (Retain Acc.), and the accuracy on the single forgotten class (Forget Acc.). The best-performing method in each subgroup is highlighted in bold, selected first by achieving a Forget Acc. ≤ 0.010, and then by having the highest Retain Acc.

Method	**Overall Acc.**	**Retain Acc.**	**Forget Acc.**
Forgetting Class 16			
Reference Models			
Original	0.729 ± 0.004	0.728 ± 0.004	0.746 ± 0.042
Retrain	0.723 ± 0.004	0.731 ± 0.004	0.000 ± 0.000
— Unlearning with Forget Data Only —			
GradAsc	0.559 ± 0.032	0.564 ± 0.032	0.018 ± 0.022
RandLabel	0.566 ± 0.014	0.572 ± 0.014	0.033 ± 0.026
Finetune	0.725 ± 0.005	0.727 ± 0.005	0.488 ± 0.050
AGE	0.519 ± 0.034	0.524 ± 0.034	0.031 ± 0.032
GHR	0.567 ± 0.037	0.573 ± 0.038	0.015 ± 0.023
GSR	0.579 ± 0.018	0.584 ± 0.018	0.014 ± 0.021
UNDIAL	0.546 ± 0.039	0.551 ± 0.039	0.027 ± 0.026
GROGU	**0.592 ± 0.011**	**0.598 ± 0.011**	**0.008 ± 0.022**
GROGU T1	0.586 ± 0.011	0.591 ± 0.011	0.022 ± 0.022
— Unlearning with Retain Data —			
GradAsc Descent	0.666 ± 0.007	0.672 ± 0.007	0.007 ± 0.018
GradRest Descent	0.666 ± 0.008	0.672 ± 0.008	0.014 ± 0.022
AGE Descent	0.679 ± 0.007	0.685 ± 0.007	0.023 ± 0.025
GHR Descent	0.680 ± 0.005	0.687 ± 0.005	0.001 ± 0.005
GSR Descent	**0.689 ± 0.005**	**0.696 ± 0.005**	**0.005 ± 0.013**
UNDIAL Descent	0.682 ± 0.005	0.689 ± 0.005	0.005 ± 0.009
GROGU Descent	0.671 ± 0.006	0.678 ± 0.006	0.001 ± 0.005
GROGU T1 Descent	0.671 ± 0.006	0.678 ± 0.006	0.009 ± 0.019
Forgetting Class 54			
Reference Models			
Original	0.729 ± 0.004	0.728 ± 0.004	0.746 ± 0.042
Retrain	0.721 ± 0.004	0.729 ± 0.004	0.000 ± 0.000
— Unlearning with Forget Data Only —			
GradAsc	0.488 ± 0.047	0.494 ± 0.047	0.012 ± 0.015
RandLabel	0.582 ± 0.014	0.588 ± 0.014	0.023 ± 0.026
Finetune	0.724 ± 0.005	0.726 ± 0.005	0.547 ± 0.049
AGE	0.531 ± 0.027	0.537 ± 0.027	0.021 ± 0.018
GHR	0.536 ± 0.049	0.542 ± 0.049	0.015 ± 0.025
GSR	0.583 ± 0.018	0.590 ± 0.018	0.010 ± 0.018
UNDIAL	0.571 ± 0.020	0.577 ± 0.021	0.013 ± 0.015
GROGU	**0.604 ± 0.007**	**0.611 ± 0.007**	**0.005 ± 0.012**
GROGU T1	0.598 ± 0.007	0.605 ± 0.007	0.021 ± 0.018
— Unlearning with Retain Data —			
GradAsc Descent	0.654 ± 0.007	0.662 ± 0.007	0.007 ± 0.012
GradRest Descent	0.654 ± 0.008	0.661 ± 0.009	0.007 ± 0.012
AGE Descent	0.677 ± 0.005	0.684 ± 0.005	0.004 ± 0.017
GHR Descent	0.672 ± 0.005	0.679 ± 0.005	0.001 ± 0.003
GSR Descent	**0.680 ± 0.004**	**0.688 ± 0.004**	**0.002 ± 0.008**
UNDIAL Descent	0.672 ± 0.005	0.680 ± 0.005	0.004 ± 0.008
GROGU Descent	0.661 ± 0.004	0.668 ± 0.004	0.000 ± 0.000
GROGU T1 Descent	0.661 ± 0.004	0.668 ± 0.004	0.001 ± 0.003
Forgetting Class 81			
Reference Models			
Original	0.729 ± 0.004	0.728 ± 0.004	0.746 ± 0.042
Retrain	0.724 ± 0.004	0.732 ± 0.004	0.000 ± 0.000
— Unlearning with Forget Data Only —			
GradAsc	0.512 ± 0.050	0.518 ± 0.050	0.011 ± 0.022
RandLabel	0.573 ± 0.018	0.579 ± 0.018	0.038 ± 0.034
Finetune	0.722 ± 0.005	0.727 ± 0.005	0.275 ± 0.075
AGE	0.551 ± 0.027	0.557 ± 0.027	0.028 ± 0.038
GHR	0.586 ± 0.029	0.592 ± 0.030	0.009 ± 0.016
GSR	0.575 ± 0.033	0.582 ± 0.033	0.015 ± 0.024
UNDIAL	0.573 ± 0.028	0.579 ± 0.028	0.030 ± 0.030
GROGU	**0.593 ± 0.008**	**0.600 ± 0.008**	**0.004 ± 0.008**
GROGU T1	0.588 ± 0.012	0.595 ± 0.012	0.033 ± 0.030
— Unlearning with Retain Data —			
GradAsc Descent	0.665 ± 0.005	0.672 ± 0.005	0.009 ± 0.026
GradRest Descent	0.662 ± 0.007	0.669 ± 0.007	0.003 ± 0.012
AGE Descent	0.680 ± 0.005	0.688 ± 0.005	0.002 ± 0.008
GHR Descent	0.676 ± 0.005	0.684 ± 0.006	0.004 ± 0.015
GSR Descent	**0.683 ± 0.005**	**0.691 ± 0.005**	**0.001 ± 0.006**
UNDIAL Descent	**0.683 ± 0.005**	**0.691 ± 0.005**	**0.005 ± 0.013**
GROGU Descent	0.661 ± 0.005	0.668 ± 0.005	0.007 ± 0.019
GROGU T1 Descent	0.659 ± 0.005	0.667 ± 0.005	0.017 ± 0.027

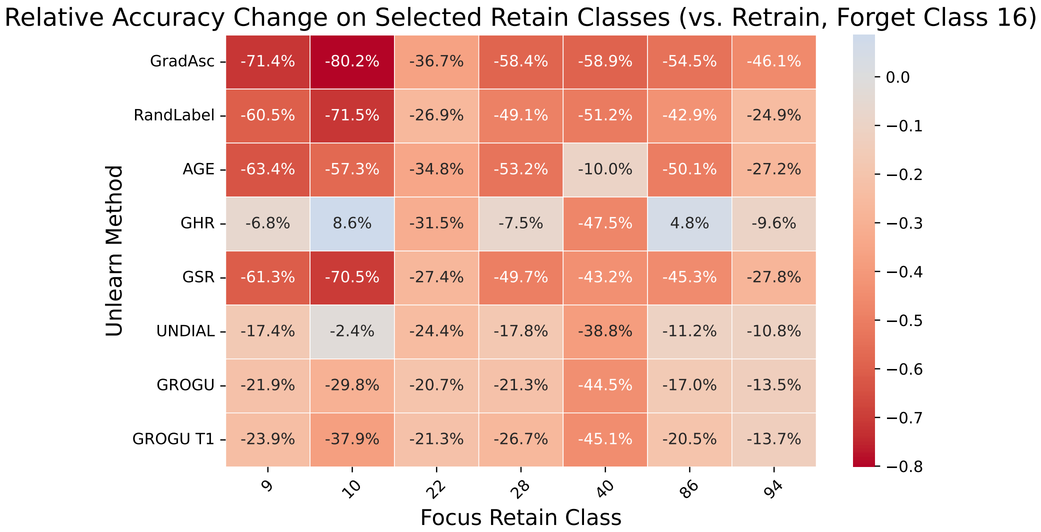


Figure 21: Heatmap of per-class accuracy drop on the CIFAR-100 retain set after unlearning class 16. Each row represents an unlearning method with forget data only, and each column corresponds to one of the relevant retain classes.

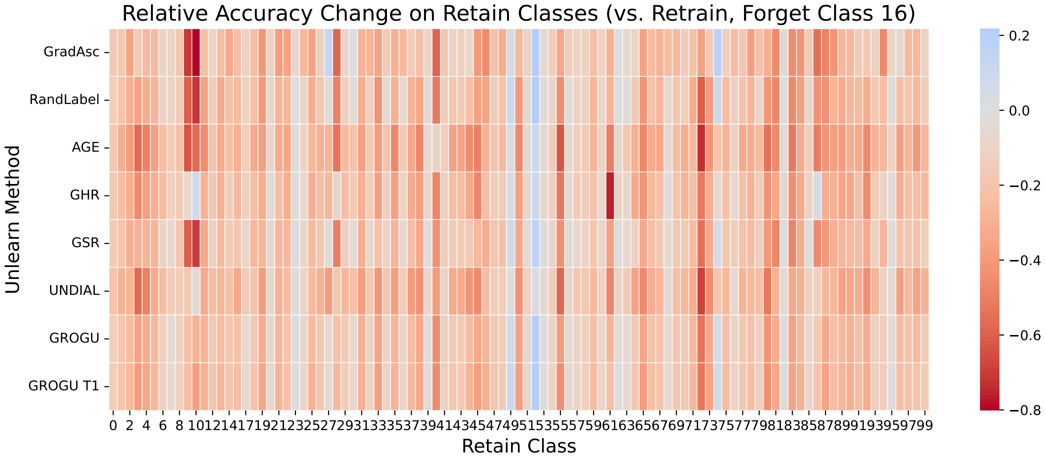


Figure 22: Heatmap of per-class accuracy drop on the CIFAR-100 retain set after unlearning class 16. Each row represents an unlearning method with forget data only, and each column corresponds to one of the 99 retain classes.

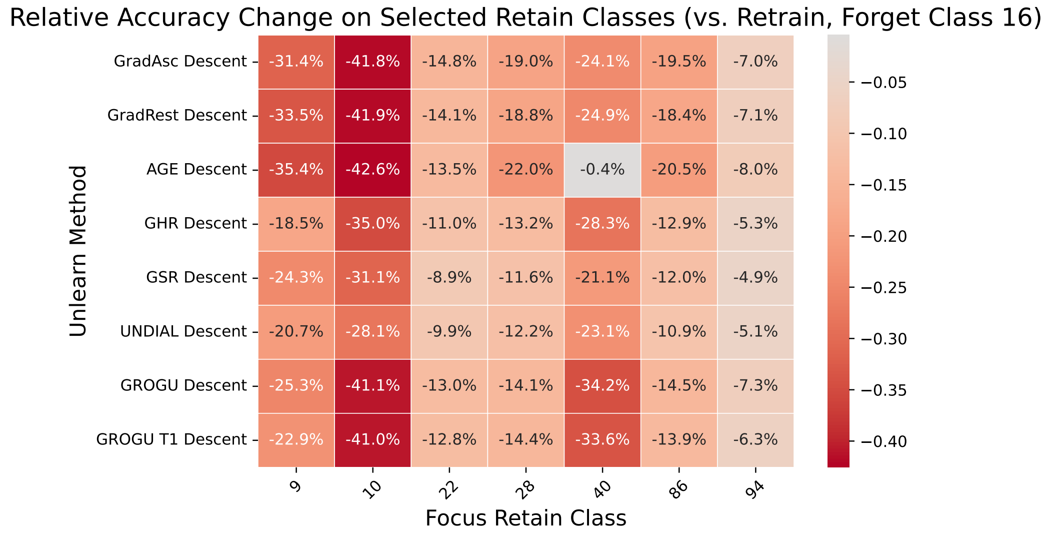


Figure 23: Heatmap of per-class accuracy drop on the CIFAR-100 retain set after unlearning class 16. Each row represents an unlearning method with forget and retain data, and each column corresponds to one of the relevant retain classes.

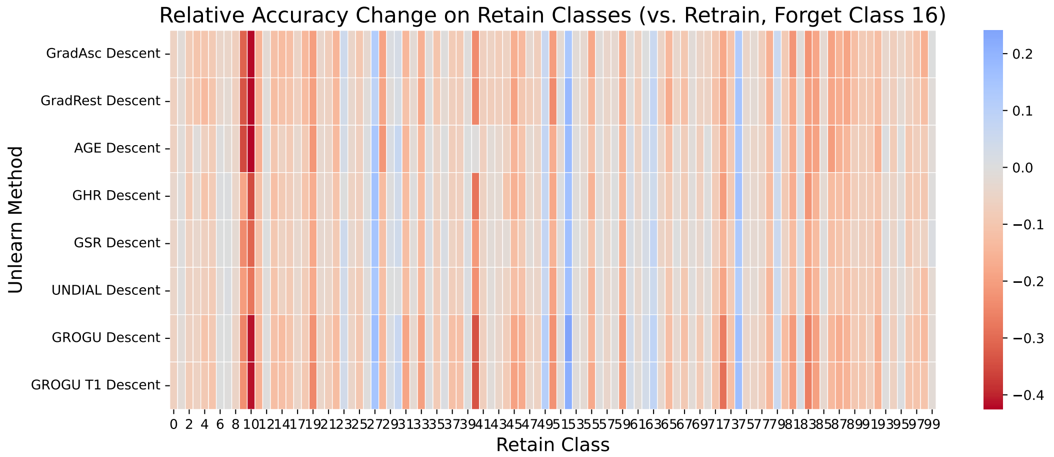


Figure 24: Heatmap of per-class accuracy drop on the CIFAR-100 retain set after unlearning class 16. Each row represents an unlearning method with forget and retain data, and each column corresponds to one of the 99 retain classes.

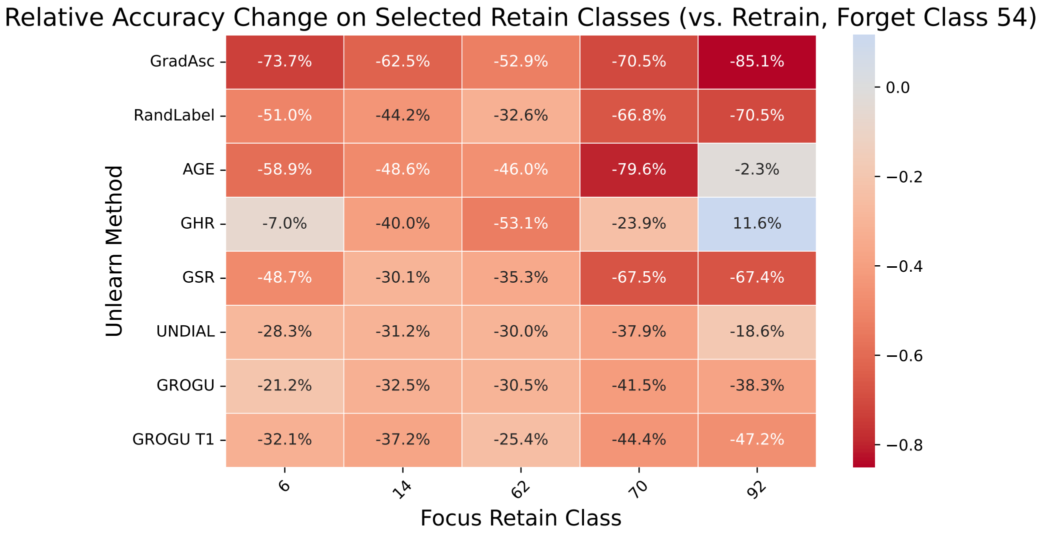


Figure 25: Heatmap of per-class accuracy drop on the CIFAR-100 retain set after unlearning class 54. Each row represents an unlearning method with forget data only, and each column corresponds to one of the relevant retain classes.

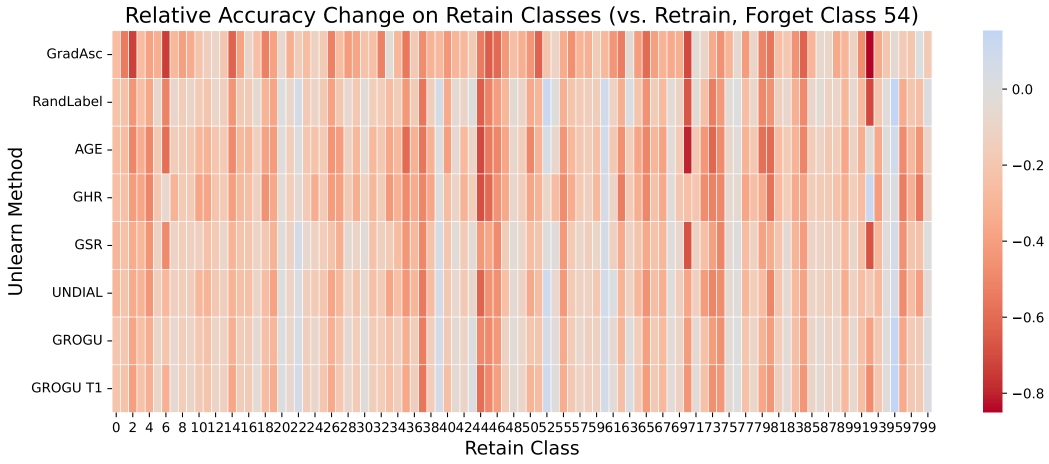


Figure 26: Heatmap of per-class accuracy drop on the CIFAR-100 retain set after unlearning class 54. Each row represents an unlearning method with forget data only, and each column corresponds to one of the 99 retain classes.

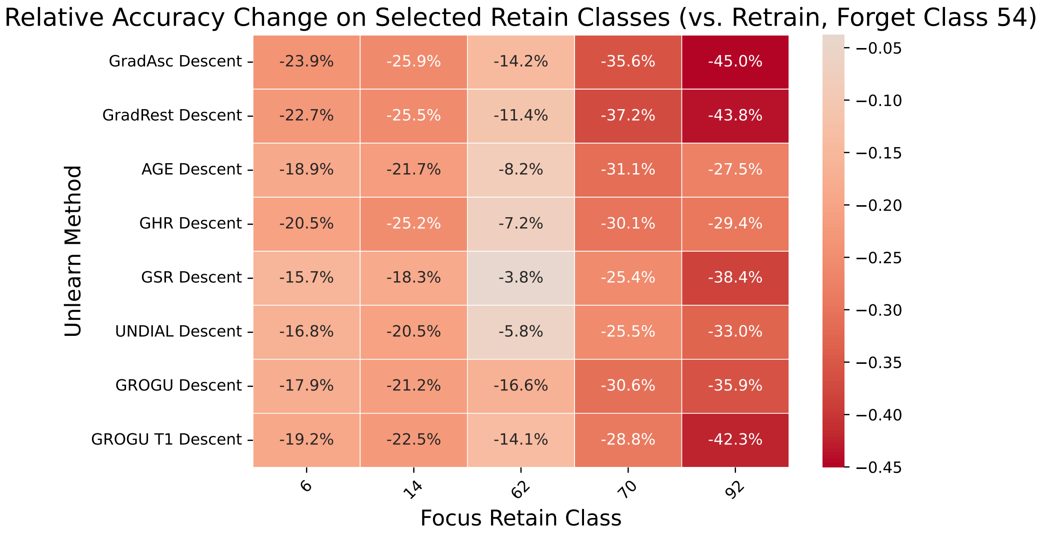


Figure 27: Heatmap of per-class accuracy drop on the CIFAR-100 retain set after unlearning class 54. Each row represents an unlearning method with forget and retain data, and each column corresponds to one of the relevant retain classes.

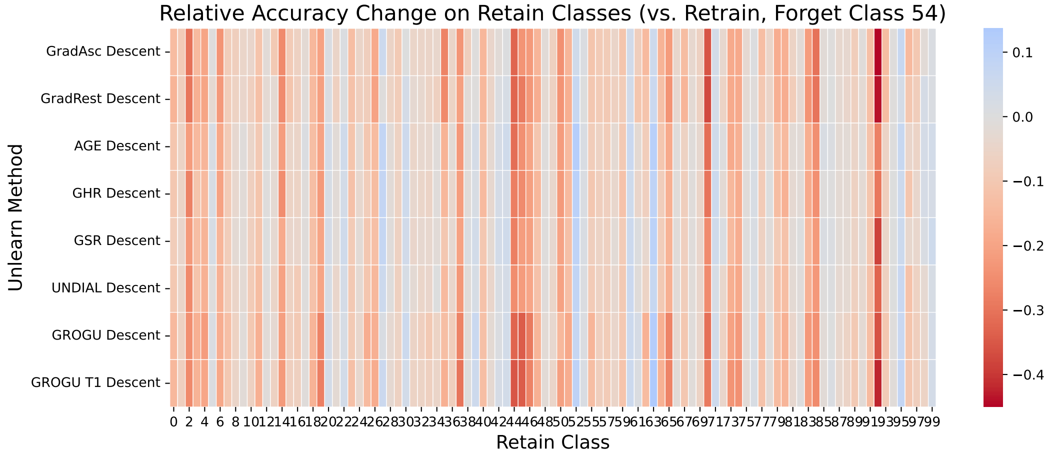


Figure 28: Heatmap of per-class accuracy drop on the CIFAR-100 retain set after unlearning class 54. Each row represents an unlearning method with forget and retain data, and each column corresponds to one of the 99 retain classes.

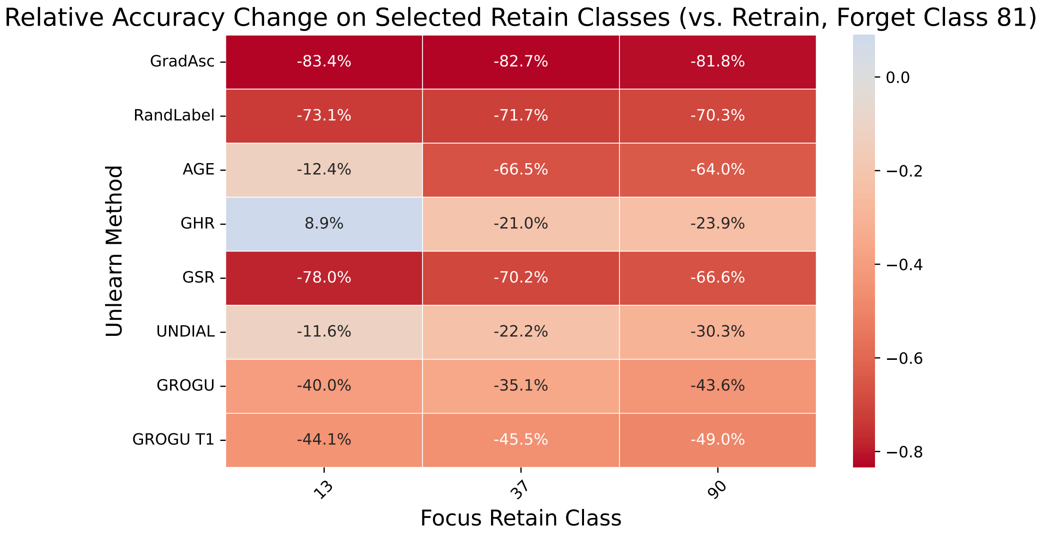


Figure 29: Heatmap of per-class accuracy drop on the CIFAR-100 retain set after unlearning class 81. Each row represents an unlearning method with forget data only, and each column corresponds to one of the relevant retain classes.

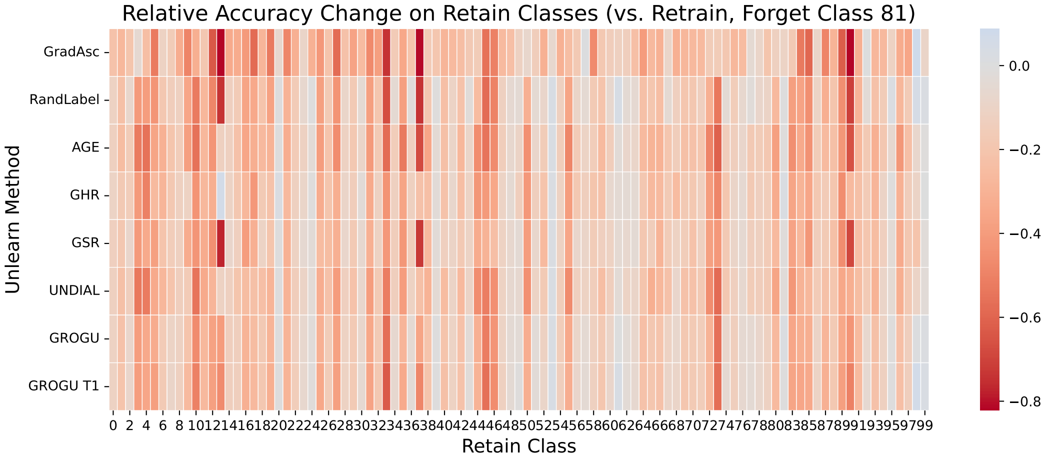


Figure 30: Heatmap of per-class accuracy drop on the CIFAR-100 retain set after unlearning class 81. Each row represents an unlearning method with forget data only, and each column corresponds to one of the 99 retain classes.

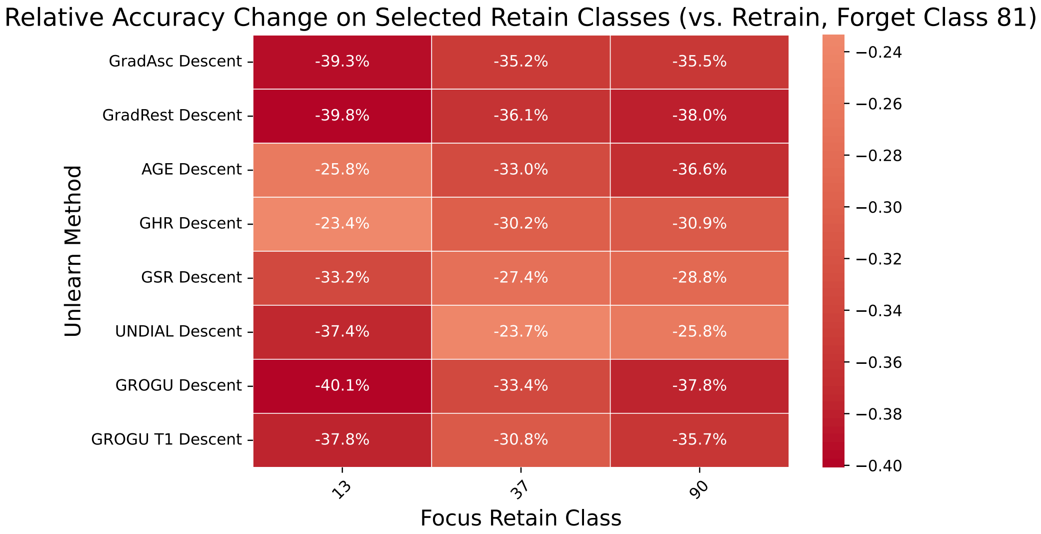


Figure 31: Heatmap of per-class accuracy drop on the CIFAR-100 retain set after unlearning class 54. Each row represents an unlearning method with forget and retain data, and each column corresponds to one of the relevant retain classes.

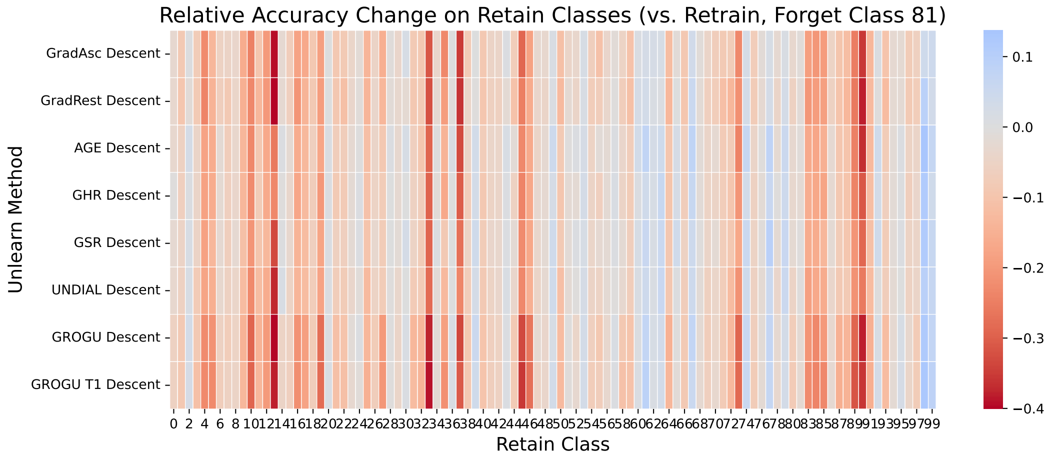


Figure 32: Heatmap of per-class accuracy drop on the CIFAR-100 retain set after unlearning class 54. Each row represents an unlearning method with forget and retain data, and each column corresponds to one of the 99 retain classes.

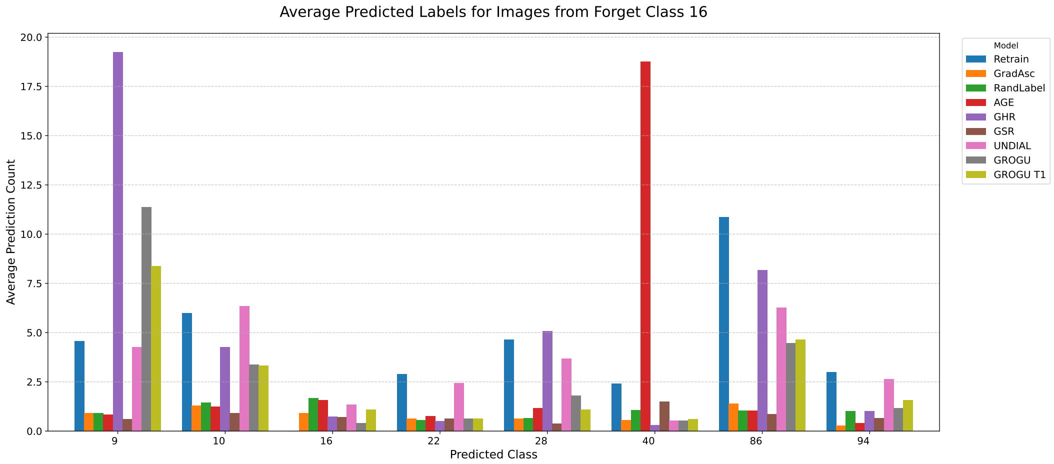


Figure 33: Prediction counts for images from the forgotten class 16 when processed by different unlearned models with forget data only. For readability, the plot focuses on the classes that the retrained model most often confused with the forgotten class.

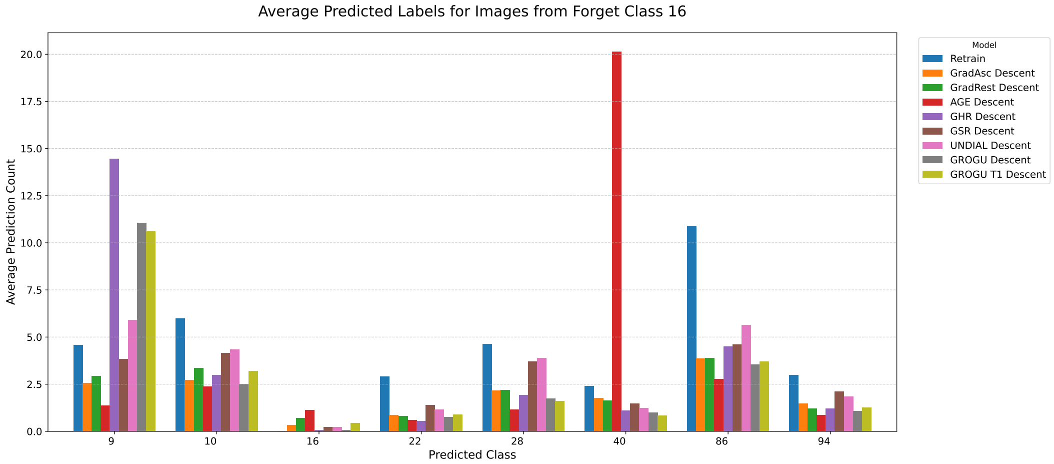


Figure 34: Prediction counts for images from the forgotten class 16 when processed by different unlearned models incorporate retain loss in addition to forget loss.

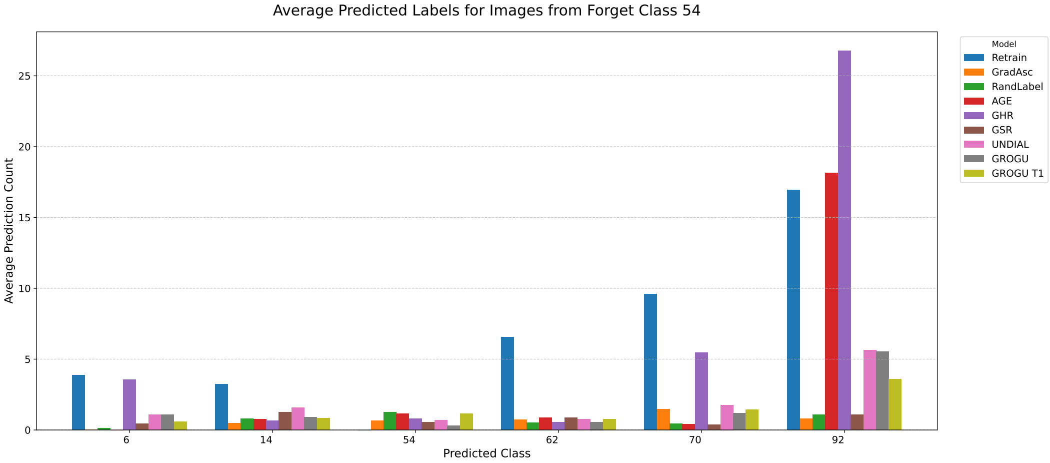


Figure 35: Prediction counts for images from the forgotten class 54 when processed by different unlearned models with forget data only.

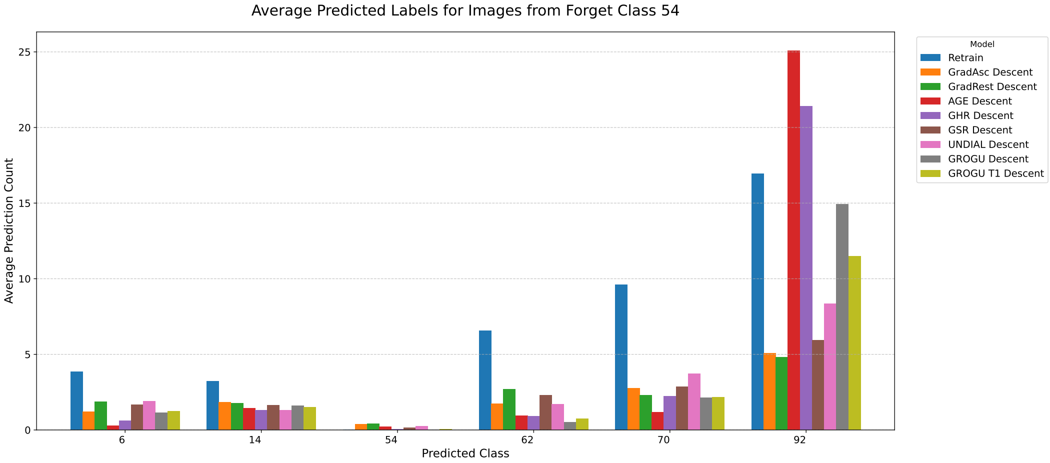


Figure 36: Prediction counts for images from the forgotten class 54 when processed by different unlearned models incorporate retain loss in addition to forget loss.

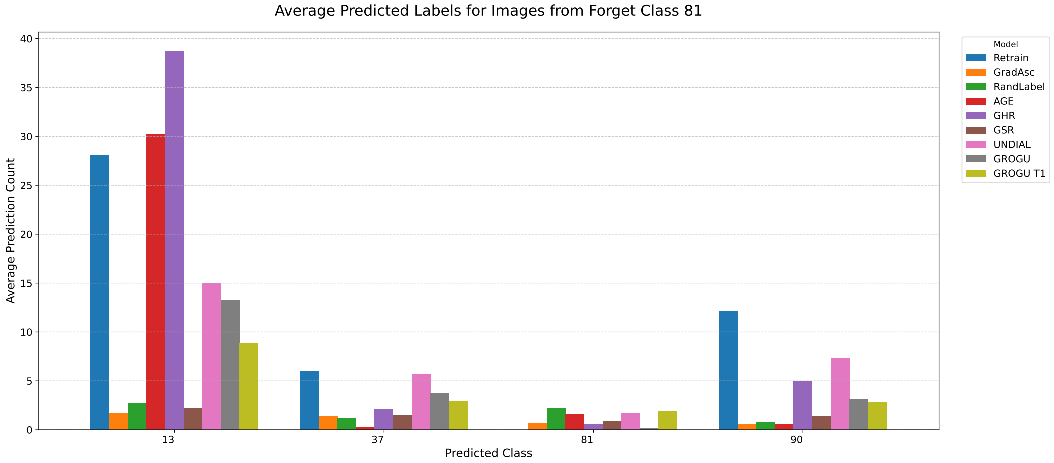


Figure 37: Prediction counts for images from the forgotten class 81 when processed by different unlearned models with forget data only.

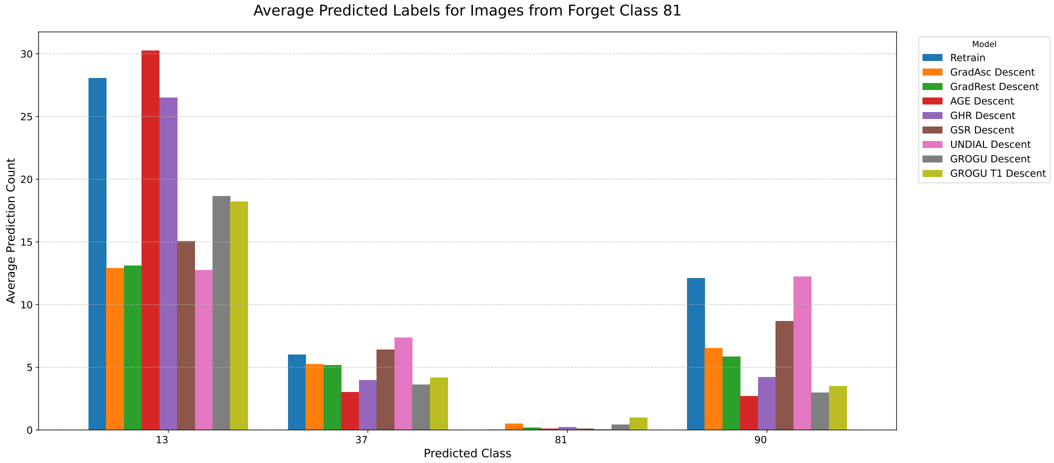


Figure 38: Prediction counts for images from the forgotten class 81 when processed by different unlearned models incorporate retain loss in addition to forget loss.

A.7 Full comparison results with baseline methods on ImageNet

This section details the results of our unlearning methods on the large-scale ImageNet dataset.

We first observe in the main results table 5 that most methods successfully achieve near-zero forget accuracy. This is largely due to the substantial number of images available for the forget class in ImageNet, which facilitates effective unlearning. It is important to note that due to time constraints, the Original and Retrain models were trained on a single seed, resulting in no standard deviation for these reference points. However, we anticipate that the variance would be minimal (≤ 0.005) across multiple seeds, similar to what is observed on CIFAR-10 datasets.

In the challenging scenario using only forget data, our GROGU method once again demonstrates superior performance, achieving the best combination of near-zero forget accuracy and the highest retain accuracy. This result validates that GROGU is exceptionally well-suited for practical applications where access to the full retain set is either prohibited or computationally infeasible.

When a limited subset of retain data is incorporated into the unlearning objective, we observe a general increase in overall accuracy across all methods. In this setting, GSR Descent emerges as a top performer. It is important, however, to contextualize this result with respect to the amount of retain data used. When a very small amount of retain data is used (e.g., 0.33% of the available subset), GROGU's advantage persists, as it is uniquely able to achieve near-zero forget accuracy while other methods fail. Conversely, with a relatively larger amount of retain data (e.g., 1% of the available subset), the unlearning task becomes less challenging, and methods like GSR Descent or UNDIAL Descent become sufficient, albeit at the cost of significantly longer computation time. For the experiments presented here, we chose a moderate amount of 0.55% to represent a realistic and challenging real-world scenario.

Further analysis of the per-class performance is provided by the accuracy drop heatmaps, where Finetune is excluded due to its high forget accuracy. In the forget-data-only setting, a focused heatmap on semantically relevant retain classes reveals that GROGU leads a much smaller accuracy drop than GSR. To visualize performance on the general retain classes, we selected the 60 classes exhibiting the most variability in accuracy drop across unlearn methods. This general-class heatmap shows that GROGU also preserves performance more effectively than GHR, UNDIAL, GradAsc, and AGE. When retain data is incorporated, we find that GSR Descent provides robust performance, preserving accuracy well across both relevant and general classes.

Table 5: Summary of performance for forget-data-only methods, representing the most challenging unlearning scenario. We report results for three scenarios (A, B, C) for each dataset, corresponding to forgetting classes {1, 3, 8} for CIFAR-10, {16, 54, 81} for CIFAR-100, and {108, 547, 892} for ImageNet. For each scenario, we report the mean test retain and forget accuracy over 30 random seeds. The best-performing approximate unlearning method is highlighted in bold, selected by first meeting a dataset-specific forget accuracy threshold ($\leq$ 0.003 for CIFAR-10, $\leq$ 0.010 for CIFAR-100, and $\leq$ 0.001 for ImageNet) and then having the highest retain accuracy.

Method	**Overall Acc.**	**Retain Acc.**	**Forget Acc.**
Forgetting Class 108			
Reference Models			
Original	0.708	0.708	0.792
Retrain	0.706	0.706	0.000
— Unlearning with Forget Data Only —			
GradAsc	0.598 ± 0.002	0.599 ± 0.002	0.000 ± 0.000
RandLabel	0.595 ± 0.014	0.596 ± 0.014	0.004 ± 0.012
Finetune	0.700 ± 0.001	0.700 ± 0.001	0.362 ± 0.037
AGE	0.500 ± 0.010	0.501 ± 0.010	0.000 ± 0.000
GHR	0.662 ± 0.001	0.662 ± 0.001	0.154 ± 0.000
GSR	0.649 ± 0.015	0.650 ± 0.015	0.015 ± 0.020
UNDIAL	0.665 ± 0.001	0.666 ± 0.001	0.077 ± 0.000
GROGU	**0.655 ± 0.001**	**0.655 ± 0.001**	**0.000 ± 0.000**
GROGU T1	0.625 ± 0.004	0.626 ± 0.004	0.000 ± 0.000
— Unlearning with Retain Data —			
GradAsc Descent	0.676 ± 0.001	0.676 ± 0.001	0.000 ± 0.000
GradRest Descent	0.675 ± 0.001	0.676 ± 0.001	0.000 ± 0.000
AGE Descent	0.649 ± 0.005	0.649 ± 0.005	0.177 ± 0.027
GHR Descent	**0.680 ± 0.001**	**0.681 ± 0.001**	**0.000 ± 0.000**
GSR Descent	0.678 ± 0.001	0.678 ± 0.001	0.023 ± 0.041
UNDIAL Descent	**0.681 ± 0.001**	**0.681 ± 0.001**	**0.000 ± 0.000**
GROGU Descent	0.672 ± 0.002	0.673 ± 0.002	0.000 ± 0.000
GROGU T1 Descent	0.660 ± 0.002	0.660 ± 0.002	0.004 ± 0.012
Forgetting Class 547			
Reference Models			
Original	0.708	0.708	0.792
Retrain	0.702	0.703	0.000
— Unlearning with Forget Data Only —			
GradAsc	0.472 ± 0.005	0.472 ± 0.005	0.000 ± 0.000
RandLabel	0.516 ± 0.015	0.517 ± 0.015	0.000 ± 0.000
Finetune	0.699 ± 0.001	0.700 ± 0.001	0.438 ± 0.096
AGE	0.554 ± 0.003	0.555 ± 0.003	0.000 ± 0.000
GHR	0.589 ± 0.002	0.589 ± 0.002	0.000 ± 0.000
GSR	0.618 ± 0.002	0.619 ± 0.002	0.000 ± 0.000
UNDIAL	0.602 ± 0.002	0.603 ± 0.002	0.000 ± 0.000
GROGU	**0.623 ± 0.001**	**0.623 ± 0.001**	**0.000 ± 0.000**
GROGU T1	0.574 ± 0.003	0.574 ± 0.003	0.000 ± 0.000
— Unlearning with Retain Data —			
GradAsc Descent	0.658 ± 0.002	0.659 ± 0.002	0.000 ± 0.000
GradRest Descent	0.658 ± 0.002	0.659 ± 0.002	0.002 ± 0.009
AGE Descent	0.662 ± 0.002	0.662 ± 0.002	0.003 ± 0.012
GHR Descent	0.664 ± 0.002	0.665 ± 0.002	0.002 ± 0.009
GSR Descent	**0.662 ± 0.002**	**0.663 ± 0.002**	**0.000 ± 0.000**
UNDIAL Descent	0.665 ± 0.002	0.666 ± 0.002	0.010 ± 0.019
GROGU Descent	0.654 ± 0.002	0.655 ± 0.002	0.000 ± 0.000
GROGU T1 Descent	0.629 ± 0.003	0.630 ± 0.003	0.024 ± 0.024
Forgetting Class 892			
Reference Models			
Original	0.708	0.708	0.792
Retrain	0.696	0.696	0.000
— Unlearning with Forget Data Only —			
GradAsc	0.647 ± 0.001	0.647 ± 0.001	0.000 ± 0.000
RandLabel	0.620 ± 0.003	0.621 ± 0.003	0.000 ± 0.000
Finetune	0.700 ± 0.002	0.701 ± 0.002	0.083 ± 0.000
AGE	0.610 ± 0.002	0.611 ± 0.002	0.000 ± 0.000
GHR	0.638 ± 0.011	0.639 ± 0.011	0.000 ± 0.000
GSR	0.670 ± 0.001	0.670 ± 0.001	0.000 ± 0.000
UNDIAL	0.640 ± 0.001	0.641 ± 0.001	0.000 ± 0.000
GROGU	**0.683 ± 0.001**	**0.684 ± 0.001**	**0.000 ± 0.000**
GROGU T1	0.644 ± 0.002	0.644 ± 0.002	0.000 ± 0.000
— Unlearning with Retain Data —			

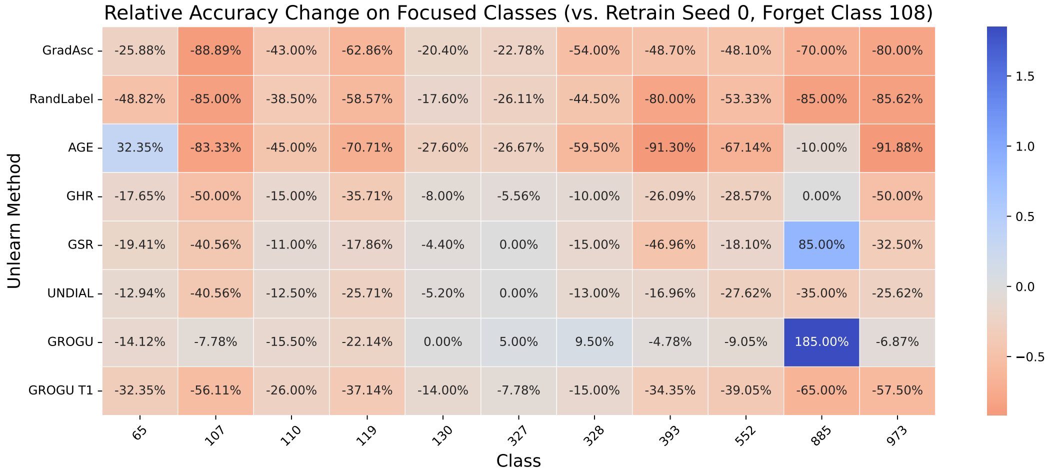


Figure 39: Heatmap of per-class accuracy drop on the ImageNet relevant retain set after unlearning class 108. Each row represents an unlearning method trained using forget data only, and each column corresponds to one of the relevant retain classes.

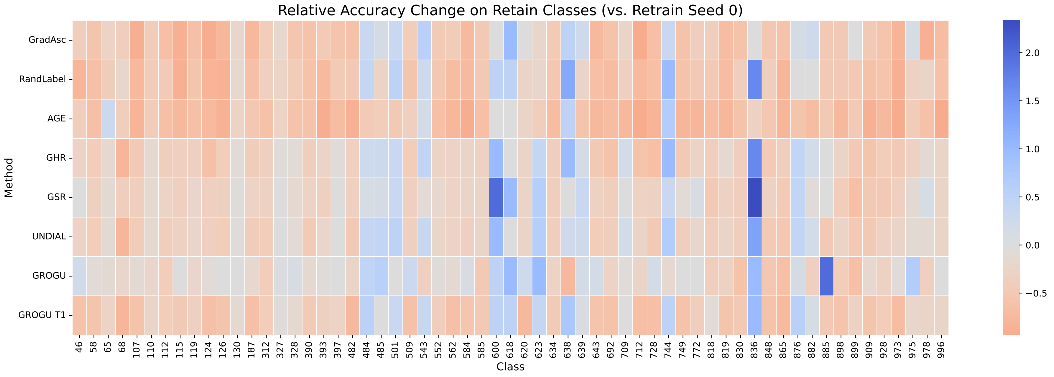


Figure 40: Heatmap of per-class accuracy drop on the ImageNet general retain set after unlearning class 108. Each row represents an unlearning method trained using forget data only, and each column corresponds to one of the general large subset of retain classes.

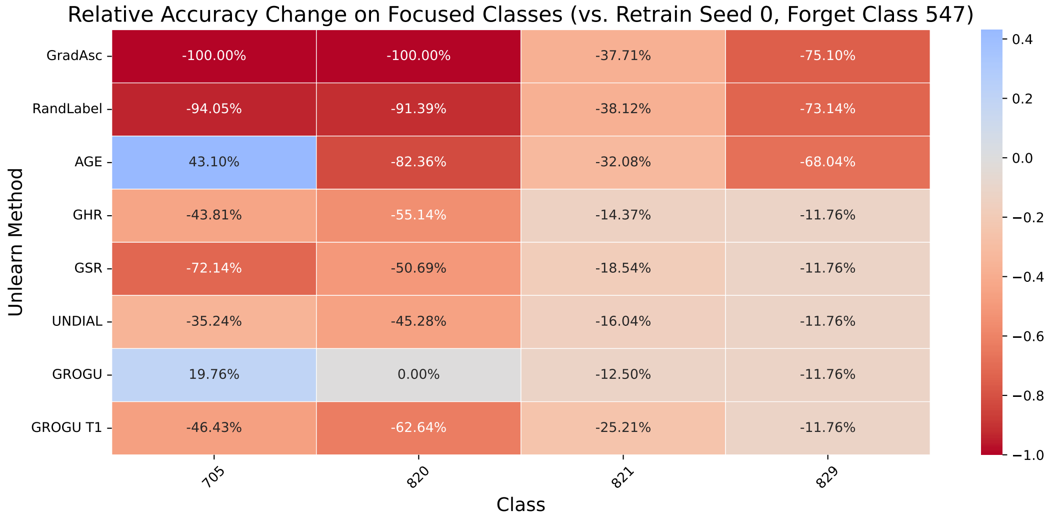


Figure 41: Heatmap of per-class accuracy drop on the ImageNet relevant retain set after unlearning class 547. Each row represents an unlearning method trained using forget data only, and each column corresponds to one of the relevant retain classes.

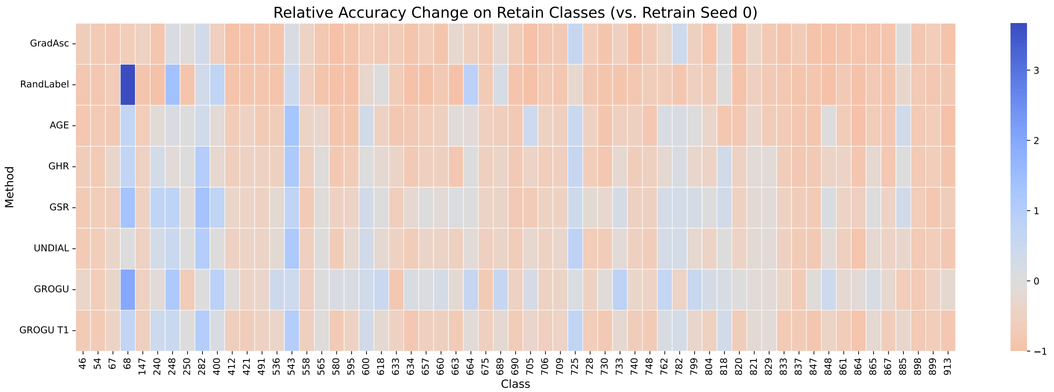


Figure 42: Heatmap of per-class accuracy drop on the ImageNet general retain set after unlearning class 547. Each row represents an unlearning method trained using forget data only, and each column corresponds to one of the general large subset of retain classes.

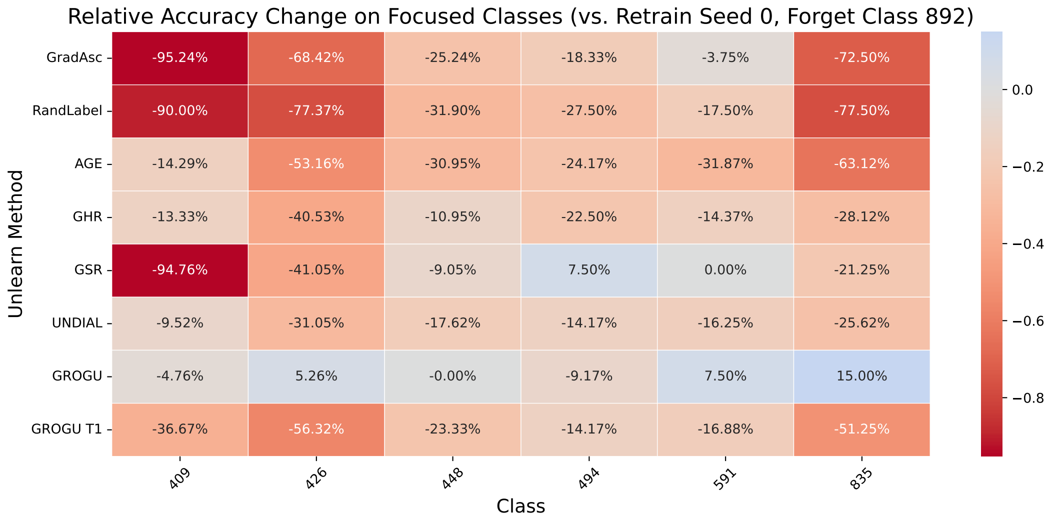


Figure 43: Heatmap of per-class accuracy drop on the ImageNet relevant retain set after unlearning class 892. Each row represents an unlearning method trained using forget data only, and each column corresponds to one of the relevant retain classes.

iclr2026/experiments_results/paper_imagenet_results/forgetclass892/accuracy_drop/plo

Figure 44: Heatmap of per-class accuracy drop on the ImageNet general retain set after unlearning class 892. Each row represents an unlearning method trained using forget data only, and each column corresponds to one of the general large subset of retain classes.

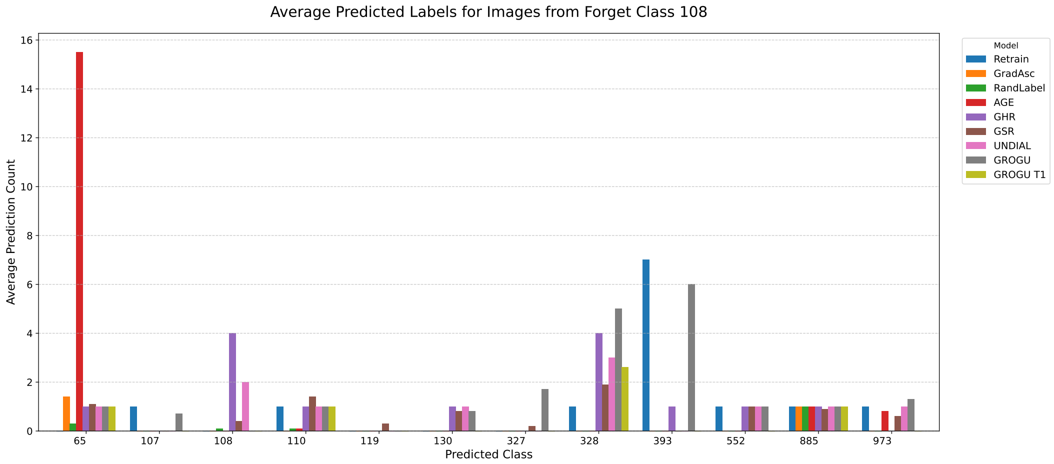


Figure 45: Prediction counts for images from the forgotten class 108 when processed by different unlearned models with forget data only. For readability, the plot focuses on the classes that the retrained model most often confused with the forgotten class.

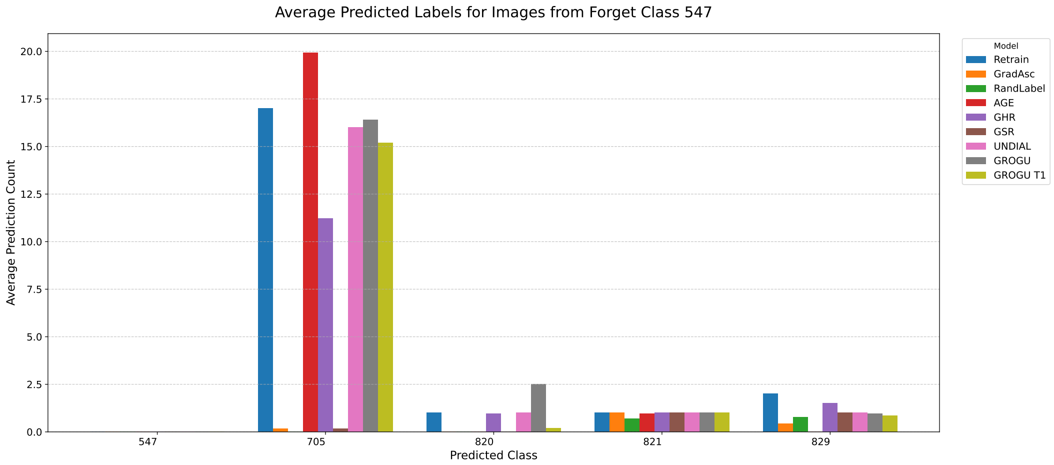


Figure 46: Prediction counts for images from the forgotten class 547 when processed by different unlearned models with forget data only. For readability, the plot focuses on the classes that the retrained model most often confused with the forgotten class.

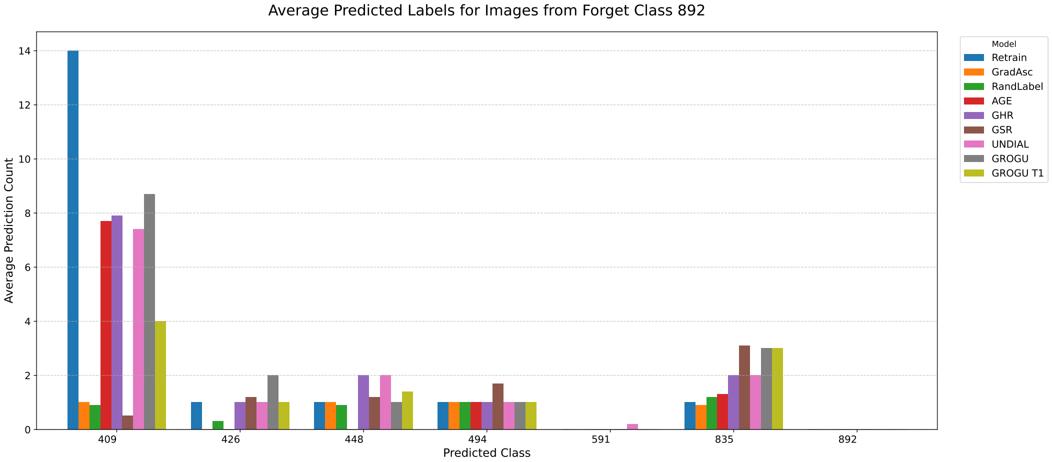


Figure 47: Prediction counts for images from the forgotten class 892 when processed by different unlearned models with forget data only. For readability, the plot focuses on the classes that the retrained model most often confused with the forgotten class.