Weight initialization based on gradient similarity for versatile machine unlearning

Doun Lee
Kookmin University
ldoun8260@gmail.com

Jongyun Shin
Kookmin University
jjongyn@gmail.com

Jinwoo Bae
Kookmin University
bgw4399@gmail.com

Hyunjoon Cho
Kookmin University
juatis96@kookmin.ac.kr

Jangho Kim*
Kookmin University
jangho.kim@kookmin.ac.kr

Abstract

The growing necessity for deep learning applications to adhere to rising data privacy standards has made machine unlearning crucial in removing the impact of specific examples from a given model. Although *exact unlearning*, which retrains the model from scratch using the remaining dataset, satisfies this objective, it is computationally expensive, leading *approximate unlearning* to be an active research field. However, we argue that most existing approximate algorithms fail to provide robust privacy guarantees. These methods are often "biased," designed to defend against attacks on a single model property (e.g., loss) while leaving information encoded in other properties (e.g., entropy) exposed. An adaptive attacker can simply exploit this bias, making the unlearning ineffective, as a model's privacy is only as strong as its most vulnerable point. For example, gradient descent using random labels may enhance indistinguishability with respect to loss by directly lowering the probability of the correct label, but it fails to achieve comparable results for entropy, as it does not increase the entropy to levels similar to those of unseen data. To address this problem, we propose a **W**eight **I**nitialization based on **G**radient similarity dubbed WIG, a novel algorithm that provides unbiased unlearning that approximates the retraining process. Instead of targeting a specific property, WIG induces catastrophic forgetting by partially initializing weights based on their gradient similarity between the train set and the data to be forgotten. WIG achieves the best indistinguishability among current state-of-the-art approximate unlearning algorithms across diverse metrics, including various membership inference attacks, Inter-class confusion test, U-LiRA, NeurIPS Machine Unlearning Challenge, and Gaussian Poison, demonstrating its versatility.

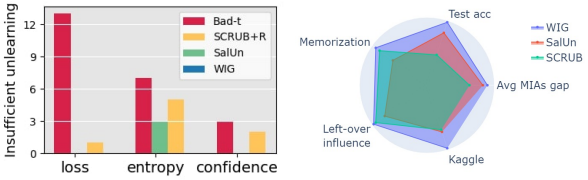
Keywords

Machine learning, Machine unlearning, Deep learning, Data privacy

1 Introduction

Machine learning (ML) enhances model performance through training data, with deep learning enabling efficient processing of large datasets, often involving sensitive information like financial and medical records. This raises privacy and copyright concerns. As a countermeasure, the EU's General Data Protection Regulation (GDPR) [17] grants individuals the 'right to be forgotten' [33]. Machine unlearning (MU) removes the influence of specific data, called the forget set, from a model. This can be accomplished by retraining with the retain set, which is the training set excluding the forget set. This process is referred to as *exact unlearning*. However, this approach is computationally expensive. Consequently, unlearning research focuses on *approximate unlearning* techniques to reduce computational costs while effectively achieving unlearning.

Membership inference attacks (MIAs) are commonly used to evaluate unlearned models by predicting the membership of the forget set. These attacks aim to differentiate model outputs between forget and retain sets, or forget and test sets, by exploiting specific properties like loss, entropy, or confidence (defined as the softmax probability of the correct class). For example, Kurmanji et al. [24] used loss-based MIA, Chundawat et al. [4] used entropy-based MIA,



(a) The number of insufficient unlearning **(b) Evaluation across different criteria**

Figure 1: (a) shows the number of insufficient unlearning experiments, which failed to defend against corresponding membership inference attacks (MIAs) targeting different model properties. We quantify insufficient unlearning by identifying algorithms that rank within the bottom 30% in unlearning performance compared to other algorithms across various experiments. Our proposed algorithm, WIG, demonstrates no instances of insufficient unlearning. Figure (b) compares SCRUB [24], SalUn [9], and WIG in 5 metrics using MIAs, Inter-class confusion test, and NeurIPS Machine Unlearning Challenge (Kaggle). Note that SCRUB+R is an extension of SCRUB. Detailed metric descriptions are available in Section 2.2.

*Corresponding author

Proceedings on Privacy Enhancing Technologies 2026(3), 575–603

https://doi.org/10.56553/popets-2026-0097

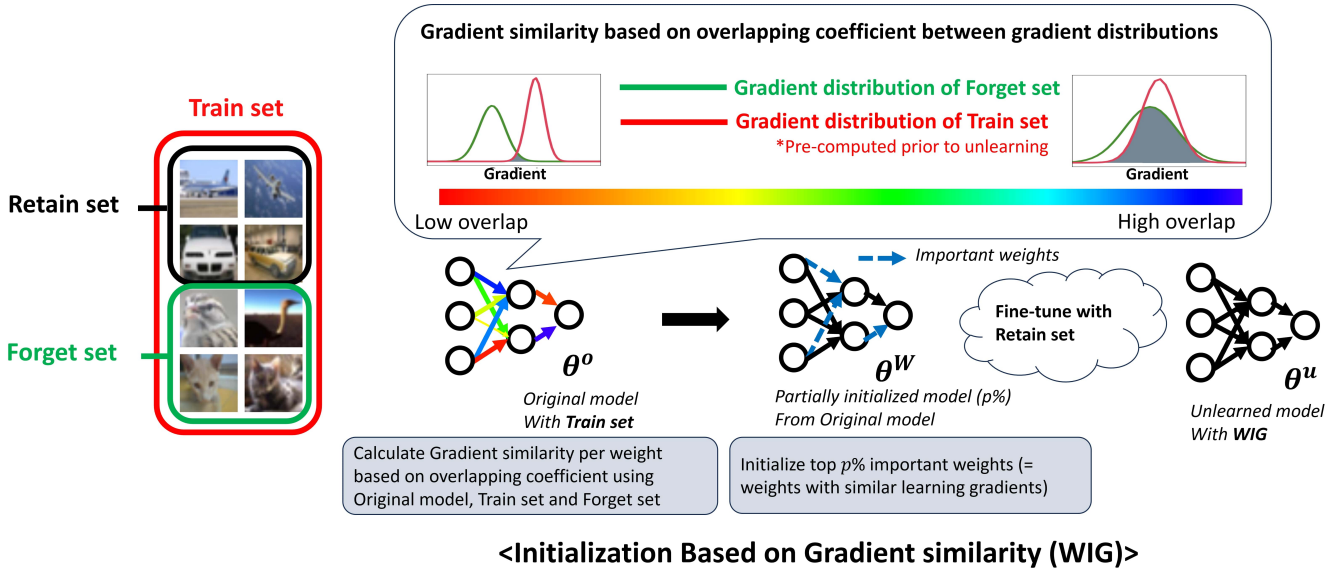


Figure 2: WIG initializes weights exhibiting high gradient similarity between the forget and retain sets (dotted blue arrows). This amplifies catastrophic forgetting during subsequent fine-tuning on the retain set, effectively approximating exact unlearning (Retrain). Our method demonstrates strong performance across diverse unlearning scenarios and metrics.

and Fan et al. [9] used confidence-based MIA to evaluate unlearning algorithms in their research. The performance of approximate unlearning is assessed by comparing the attack success rate of exact and approximate unlearning. A small gap indicates that the attacker cannot distinguish between the retrained and unlearned models based on their outputs, resulting in a stronger defense. However, existing approximate unlearning algorithms, including state-of-the-art algorithms such as Bad-T [4], SCRUB+R [24], and SalUn [9] fail to demonstrate robust performance under different MIAs that leverage various model properties. This is evident from Figure 1a, which shows the number of unlearning experiments that failed to reduce the MIAs gap compared to other algorithms across various forgetting scenarios, models, and datasets. We counted the number of methods ranked in the bottom 30% in terms of the MIA gap between unlearned and retrained models for various model properties compared to other unlearning algorithms. We hypothesized that existing algorithms might be over-fitted at unlearning certain properties of the model, given the fact that most successful MIA for SCRUB+R and SalUn was based on loss and confidence, respectively, which are the MIAs that Kurmanji et al. [24], and Fan et al. [9] used to evaluate their algorithms in their research.

The varying unlearning performance across different model properties reflects the inherent bias in the different forgetting procedures. For example, one can unlearn by directly increasing the loss of the forget set by using gradient ascent (SCRUB+R [24]), increasing the entropy of the forget set by distilling random noise (Bad-T [4]) or using gradient descent with a random label for the forget set (SalUn [9]), which is shown in the abstract. These are all biased to target specific properties of the model while showing weakness in others, which also aligns with the most unsuccessfully unlearned model properties for each algorithm in Figure 1a. Therefore, we argue that most existing approximate unlearning algorithms fail to provide robust privacy guarantees. They are often designed to defend against an attack on a single model property while leaving information encoded in other properties exposed. An attacker, however, is not constrained to a single, pre-defined attack. Being adaptive, they will perform attack on the model's least defended property.

To address these issues, we propose a **W**eight **I**nitialization based on **G**radient similarity (WIG), an approximate unlearning algorithm highly influenced by continual learning that incorporates catastrophic forgetting and initialization. In Neyshabur et al. [30], from the perspective of fine-tuning, models initialized from the same pre-trained model demonstrate a flat basin in the loss landscape and perform a consistent performance. On the other hand, models initialized randomly did not demonstrate such performance. We partially initialize the model and fine-tune using the retain set from these insights to achieve forgetting and faster convergence. Unlike the previous approach, this forgetting procedure does not directly use the forget set for unlearning. This key difference makes it unbiased to specific model properties, a significant advantage demonstrated by the zero instances of insufficient unlearning shown in Figure 1a. This procedure can also be seen as an approximation of retraining because our procedure is equivalent to retraining in that it initializes the model and then fine-tunes with the retain set, where we approximate retraining by partial initialization and shorter fine-tuning.

Additionally, we present weight selection via gradient similarity for weight initialization to maximize catastrophic forgetting during later fine-tuning to remove the influence of the forget set. In continual learning, where different tasks are learned consecutively, catastrophic forgetting inevitably arises. Lopez-Paz and Ranzato

[26] circumvented catastrophic forgetting by imposing a regularization condition that ensures the similarity of learning gradients between different tasks is greater than zero. Inspired by these findings, we initialized weights named important weights with similar learning gradients between the forget and train sets to induce catastrophic forgetting of the forget set's learning patterns. This is because high gradient similarity implies the presence of shared features. By identifying and surgically initializing these weights, we effectively remove the components that would demagnify the catastrophic forgetting. The overall process of WIG is depicted in Figure 2. Throughout our paper, we empirically show that WIG achieves the best indistinguishability across diverse metrics, unlearning scenarios, models, and datasets, as displayed in Figure 1b. Our main contributions are summarized as follows:

• We demonstrate that state-of-the-art approximate unlearning algorithms provide biased privacy defenses that are vulnerable to an adaptive attacker, and we argue this "least defended property" is a critical privacy failure.

• We propose WIG, which approximates retraining by partially initializing weights, while weight selection via gradient similarity maximizes catastrophic forgetting during later fine-tuning.

• We demonstrate that WIG outperforms existing unlearning algorithms across diverse unlearning scenarios and metrics.

2 Problem definition and forgetting metrics

2.1 Problem definition

Machine unlearning dataset: We define $\mathcal{D} = \{x_i, y_i\}_{i=1}^{N}$ to be a train set consisting of N samples, and $\mathcal{D}^{Te} = \{x_i, y_i\}_{i=1}^{M}$ to be a test set consisting of M samples, with i-th sample and its label as x_i and y_i, respectively. Let $\mathcal{D}^F$ be the forget set whose influence we want to erase from the model, which is a subset of $\mathcal{D}$. $\mathcal{D}^F$ can be composed of single or multiple classes, which contains the data we wish to remove. Similarly, $\mathcal{D}^R$ is defined as a retain set that needs to be retained. Therefore, our forget set $\mathcal{D}^F$ and retain set $\mathcal{D}^R$ satisfy $\mathcal{D}^F \cup \mathcal{D}^R = \mathcal{D}$ and $\mathcal{D}^F \cap \mathcal{D}^R = \emptyset$.

Notations: Consider the cross-entropy loss as $\mathcal{H}(y_i, \hat{y}_i \mid \theta) = -\sum_{c=1}^{C} p(y_i(c)) \log p(\hat{y}_i(c) \mid \theta)$ for C classes, where y_i is the ground truth for the i-th instance, and $\hat{y}_i$ is the predicted output by the model θ. Then, the empirical risk minimization (ERM) loss function is defined as:

$$\mathcal{L}(\theta; \text{Data Set}) = \frac{1}{|\text{Data Set}|} \sum_{i=1}^{|\text{Data Set}|} \mathcal{H}(y_i, \hat{y}_i \mid \theta) \tag{1}$$

Machine unlearning models: We define the original model trained on the train set $\mathcal{D}$ as $\theta^o = \arg\min_\theta \mathcal{L}(\theta; \mathcal{D})$. The model trained from scratch on retain set $\mathcal{D}^R$ is defined as *gold model*($\theta^{Gold} = \arg\min_\theta \mathcal{L}(\theta; \mathcal{D}^R)$) also known as *exact unlearning*. Our goal is to find an *unlearned model* (θ^u) that is similar to *gold model* by updating *original model* ($\theta^o \rightarrow \theta^u$) with an *approximate unlearning*.

Unlearning scenarios: We consider three scenarios for constructing a forget set $\mathcal{D}^F$ to remove information that should be forgotten. **(i) Random forgetting** randomly selects 10% of data from $\mathcal{D}$. **(ii) Class forgetting** is a subset of $\mathcal{D}$ that consists solely of a single class. **(iii) Selective forgetting** consists of 100 examples randomly selected from a single class.

2.2 Forgetting metrics

Membership Inference Attacks (MIAs): MIAs are widely used to evaluate the indistinguishability between exact unlearning and approximate unlearning of the forget set $\mathcal{D}^F$. By formulating a binary attack classification, the attack model is trained to predict the membership of an example based on the model's property. The attack success rate (ASR) indicates the rate of $\mathcal{D}^F$ classified as a member of $\mathcal{D}$. Then, indistinguishability is measured by the gap between the ASR of θ^u and θ^{Gold}. Our work evaluates the algorithms using various MIAs previously developed for different model properties. Specifically, we leverage techniques based on model loss [24], entropy [13], and confidence [19] to train and evaluate our attack model, named MIA loss, MIA entropy, and MIA confidence throughout this paper. These MIAs differ in their training and testing stages. MIA loss is trained on a class-balanced dataset sampled from the loss of $\mathcal{D}^F$ and $\mathcal{D}^{Te}$. In contrast, MIA entropy and MIA confidence are trained on a class-balanced dataset sampled from $\mathcal{D}^R$ and $\mathcal{D}^{Te}$. Due to these differences in the training stage, the exact unlearning (θ^{Gold}) should ideally achieve an ASR close to 50% for MIA loss, as $\mathcal{D}^F$ and $\mathcal{D}^{Te}$ are simply two different held-out datasets from its perspective. During the testing stage, the attack model is evaluated on the remaining subset of $\mathcal{D}^F$ and $\mathcal{D}^{Te}$ for MIA loss, while MIA entropy and MIA confidence are evaluated on the entire $\mathcal{D}^F$.

Accuracy of $\mathcal{D}^F$ & $\mathcal{D}^R$ & $\mathcal{D}^{Te}$: Accuracy of the forget, retain, and test sets are defined as accuracy of θ^u on $\mathcal{D}^F$, D^R, and $\mathcal{D}^{Te}$, respectively. In the class forgetting scenario, the class to be forgotten is excluded from $\mathcal{D}^{Te}$. In approximate unlearning, we aim to create a model that behaves similarly to exact unlearning. Therefore, a good approximate unlearning algorithm should reduce the accuracy gap on $\mathcal{D}^F$ compared to exact unlearning. Also, throughout the process of unlearning, the model's utility (i.e., the accuracy of $\mathcal{D}^R$ and $\mathcal{D}^{Te}$) should not be damaged as much as possible.

Average gap: If unlearning was successful, the unlearned model should show small gaps in all metrics, achieving a value close to zero when averaged. To partially satisfy this criterion, we average the gap between θ^u and θ^{Gold} across MIA loss, MIA entropy, MIA confidence, and accuracy on $\mathcal{D}^F$.

Inter-class confusion test: A successful unlearning should remove both memorization of $\mathcal{D}^F$ and the influence of $\mathcal{D}^F$ on the unseen data. Therefore, we employ the Inter-class confusion test proposed by Goel et al. [11] to fully capture the indistinguishability. This test works by adversarially manipulating data for training (by swapping labels between two classes) and then unlearning this manipulated data. If unlearning was successful, the influence of manipulation should be resolved both in $\mathcal{D}^F$ and the unseen data. The core idea of this evaluation is to measure the number of misclassifications that only occurred between two selected classes, called targeted error. Specifically, if C^D is the confusion matrix for dataset D, and $C^D_{A,B}$ is the number of examples from class A misclassified as class B, the targeted errors are formulated as:

$$\text{Targeted Err}(D) = C^D_{A,B} + C^D_{B,A} \tag{2}$$

We can interpret the targeted error of $\mathcal{D}^F$ as the memorization of $\mathcal{D}^F$ and the targeted error of $\mathcal{D}^{Te}$ as the left-over influence of $\mathcal{D}^F$

in the unseen data. So, a successful unlearning algorithm should reduce the targeted errors in both datasets while preserving the accuracy on $\mathcal{D}^{Te}$ as much as possible.

U-LiRA. U-LiRA[14] works by adapting the LiRA[3] to evaluate unlearning algorithms on a per-example basis, offering a significantly stronger assessment than standard MIAs. The method employs shadow models to simulate scenarios where specific data points were either unlearned or never seen. It then calculates a likelihood ratio to quantify any retained information about the target example. A method that reduces membership inference accuracy to random guessing by simulating the distribution of unseen data is considered effective.

NeurIPS machine unlearning challenge: Machine unlearning challenge [37] was held on Kaggle. Given the ResNet-18 [15] classifier trained to classify the age group of natural images of people's faces [40] for 30 epochs, its objective was to develop an algorithm that can forget a subset of examples. The unlearning efficacy of the participant's algorithms is evaluated by measuring the distribution difference F between multiple θ^u and θ^{Gold} for the different forget samples. This difference is calculated using the most successful attack performed during M threshold attacks, based on the confidence of multiple θ^u and θ^{Gold} for each example in $\mathcal{D}^F$. The final score is determined by multiplying the distribution difference F by the model utility, which is defined as the accuracy of θ^u over θ^{Gold} on the $\mathcal{D}^R$ and $\mathcal{D}^{Te}$, formulated as:

$$\text{Score} = F \times \frac{Acc(D^R, \theta^u)}{Acc(D^R, \theta^{Gold})} \times \frac{Acc(D^{Te}, \theta^u)}{Acc(D^{Te}, \theta^{Gold})} \tag{3}$$

Also, note that this algorithm uses the LiRA[3] approximated attack when measuring the distribution difference. It approximates it by using a single original model θ^o, whereas LiRA uses multiple original models for evaluation.

Gaussian poison. Gaussian poison[31] injects undetectable noise into training samples to induce memorized gradient traces. An auditor detects these traces using the Gaussian Unlearning Score calculated as the normalized inner product between the input gradient and the noise vector. While effective unlearning yields a standard normal distribution, memorization causes a distributional shift quantified by the True Positive Rate (TPR) at a low False Positive Rate (FPR) of 1%. A high TPR at this threshold indicates the reliable detection of $\mathcal{D}^F$ information and the failure of the unlearning process.

2.3 Privacy goal

Attacker's goal: The adversary's goal is to determine if a specific forget set $\mathcal{D}^F$ was used to train a given model. They do this by attempting to distinguish the θ^u from θ^{Gold}. **Attacker's capabilities:** We assume an adaptive, black-box attacker. The attacker can query the model with data from the forget set $\mathcal{D}^F$ and observe its outputs. Critically, the attacker is adaptive and not limited to a single model property. They will leverage the diverse metrics discussed in Section 2.2 (e.g., loss, entropy, and confidence) to find the model property that reveals the most information. **Privacy goal:** Our objective is to develop an unlearning algorithm that produces a model θ^u that is indistinguishable from θ^{Gold}. This indistinguishability must hold across all model properties an adaptive attacker might query. An unlearning method that fails to defend even one property has failed to achieve this privacy goal.

3 Related works

Machine unlearning. Machine unlearning is a concept proposed by Cao and Yang [2] that focuses on implementing forgetting systems to remove the influence of specific data from the model trained by the machine learning process. **Retrain** refers to training the model from scratch using only $\mathcal{D}^R$. However, due to its high costs in both training and computation, *approximate unlearning* for efficient unlearning has been emerging as an alternative recently. **Fine-tuning** finetunes the model with $\mathcal{D}^R$. **NegGrad** updates the model by applying gradient ascent with $\mathcal{D}^F$ and gradient descent with $\mathcal{D}^R$. **GA** performs gradient ascent optimization on $\mathcal{D}^F$. **Fisher Forgetting (FF)** [12] proposes a scrubbing method to remove information by perturbing the model weights using the Fisher Information Matrix [28]. **Influence Unlearning (IU)** utilizes an influence function [5] approach to estimate model parameter changes during training. **Exact-unlearning Forgetting-k (EUK)** and **Catastrophic Forgetting-k (CFK)** [11] suggest unlearning strategies for the last K layers from fine-tuning from scratch or θ^o, respectively, while keeping all previous layers fixed. **Sparse** [19] proposes a sparsity-aware unlearning framework that utilizes weight pruning to reduce the gap between *approximate unlearning* and *exact unlearning*. **SalUn** [9] proposes a gradient-based weight saliency map found by hard thresholding on the gradient of the forget loss. After finding the map, it trains the model using $\mathcal{D}^R$ fine-tuning and $\mathcal{D}^F$ random label fine-tuning while only updating the weights the map selects. For generation models, random label fine-tuning was modified by adding a hyperparameter C, which allows the user to select class C to mislead $\mathcal{D}^F$ during the training of the generation model. **Bad-T** [4] is an approach that utilizes the teacher-student framework. In this method, $\mathcal{D}^F$ information is unlearned using an incompetent teacher, while a competent teacher is leveraged to minimize the performance drop during unlearning. **SCRUB** [24] iteratively updates "min-step" on $\mathcal{D}^R$ to prevent performance degradation while performing "max-step" to remove information about $\mathcal{D}^F$. As an extension of that, **SCRUB+R** [24] uses a validation set with the same distribution as $\mathcal{D}^F$ to determine points with moderately high errors for $\mathcal{D}^F$. **SSD** [10] selectively dampens parameters that are specifically important to $\mathcal{D}^F$ by leveraging the Fisher information matrix of $\mathcal{D}^F$ and $\mathcal{D}$. **Scissorhands** [38] takes a similar approach to ours. First, it partially initializes weights with the largest connection sensitivity to the $\mathcal{D}^F$. The method subsequently optimizes the partially initialized model in the direction that increases $\mathcal{D}^F$ loss while minimizing $\mathcal{D}^R$ loss. While Scissorhands amplifies catastrophic forgetting via gradient projection, WIG induces this effect through targeted partial initialization. **DeepClean** [34] initializes weights with high Fisher information on $\mathcal{D}^F$ and low information on $\mathcal{D}^R$ to zero before selectively training them while freezing other parameters. Unlike DeepClean which focuses on $\mathcal{D}^F$ importance, WIG identifies weights that magnify catastrophic forgetting upon initialization. Section 5.3 verifies that such importance-based selection can unintentionally reduce the effect of catastrophic forgetting

as evidenced by an upward average gap trend during subsequent fine-tuning.

4 Proposed method

Continual learning research addresses catastrophic forgetting using regularization methods such as [1, 20, 26, 39]. These methods selectively slow down the learning of weights deemed important for previously learned tasks, thereby mitigating the catastrophic forgetting of those prior tasks. This importance is typically assessed by analyzing gradients from prior tasks or by measuring the similarity of gradients between the prior and the new task. This indicates that information about the prior tasks is embedded in the selected weights, and the data it was trained on might still remain retrievable from the weights, which aligns with the necessity of machine unlearning. Moreover, continual learning relies on features shared across tasks to maintain performance [8, 22, 41]. Therefore, identifying and initializing weights that encode shared features of $\mathcal{D}^R$ and $\mathcal{D}^F$ is important for magnifying catastrophic forgetting. Inspired by this line of research, we conversely initialize important weights of θ^o based on gradient similarity to induce catastrophic forgetting of $\mathcal{D}^F$. By using gradient similarity, we effectively identify weights that encode shared features between $\mathcal{D}^R$ and $\mathcal{D}^F$. After initialization, fine-tuning with $\mathcal{D}^R$ repairs the utility damage caused by initialization while inducing catastrophic forgetting. The proposed approximate unlearning method is termed Weight Initialization based on Gradient similarity (WIG). In the following sections, first, we define gradient similarity based on two Gaussian distributions. Second, we propose a layer-wise initialization method for important weights selected by gradient similarity. Third, we show the benefits of replacing $\mathcal{D}^R$ with $\mathcal{D}$ while measuring gradient similarity. Lastly, we analyze the effect of partial initialization in terms of the difference in loss value from θ^{Gold}.

4.1 Gradient similarity based on overlapping coefficient between Gaussian distributions

In our approach, starting from θ^o, we train solely on $\mathcal{D}^R$ to induce catastrophic forgetting. However, training on $\mathcal{D}^R$ may still incorporate influences from $\mathcal{D}^F$, since $\mathcal{D}^F$ and $\mathcal{D}^R$ contain shared features. Therefore, initializing these weights effectively removes information from $\mathcal{D}^F$ during later fine-tuning.

To find these weights, we must compare the gradient distributions of $\mathcal{D}^R$ and $\mathcal{D}^F$. The most theoretically robust method would be a non-parametric test, which makes no assumptions about the data's form. However, common non-parametric statistics like the Kolmogorov-Smirnov (K-S) statistic require processing and sorting all gradient samples. For modern deep networks, this approach is practically infeasible due to computational and memory constraints, leading to out-of-memory (OOM) errors and impractical processing times, as we confirm in our ablation study (Section 5.3, Figure 8).

To design a scalable and practical algorithm, we propose a more efficient proxy. We model two distinct gradient distributions for each model weight (one for $\mathcal{D}^R$ and one for $\mathcal{D}^F$) as Gaussian. This parametric assumption allows for a highly efficient summary of each distribution using only its mean and variance.

Following Inman and Bradley Jr [18], we define the similarity between these two probability distributions using the overlapping coefficient (OVL). We designate weights with significant overlap between these two distributions as important weights. This design choice using OVL on Gaussian modeled distributions provides a fast and scalable metric that serves as an effective proxy sufficient for ranking weight importance. Weights exhibiting high OVL values encode shared features between $\mathcal{D}^R$ and $\mathcal{D}^F$. Consequently, optimizing the model with these weights using solely $\mathcal{D}^R$ may inadvertently preserve information relevant to $\mathcal{D}^F$.

Theoretical Motivation: Let w be a weight multiplying an input activation a. The gradient is given by $\frac{\partial \ell}{\partial w} = \delta a$, where δ is the error signal. Consequently, the gradient distribution depends on both a and δ. If $\mathrm{OVL}(g_{\mathcal{D}^R}, g_{\mathcal{D}^F}) \approx 1$, the gradients on both datasets are nearly indistinguishable, which suggests that w processes a shared feature pathway with similar activation patterns. Initializing such parameters disrupts this shared representation and effectively amplifies catastrophic forgetting of $\mathcal{D}^F$. Therefore, we consider weights with large OVL to be important weights and initialize them accordingly. Considering μ_1, σ_1 represent the mean and standard deviation of gradients calculated from $\mathcal{D}^R$ and μ_2, σ_2 denote the mean and standard deviation of gradients calculated from $\mathcal{D}^F$, we can compute A and B, the intersection points of the two Gaussian distributions as follows:

$$\begin{aligned} A, B &= \frac{\mu_1\sigma_2^2 - \mu_2\sigma_1^2 \pm \sigma_1\sigma_2 T^{\frac{1}{2}}}{\sigma_2^2 - \sigma_1^2}, \\ \text{where } T &= (\mu_1 - \mu_2)^2 + (\sigma_2^2 - \sigma_1^2)\ln\frac{\sigma_2^2}{\sigma_1^2} \end{aligned} \tag{4}$$

$$\begin{aligned} \Phi_l^z = \phi(\frac{A_l^z - \mu_1}{\sigma_1}) + \phi(\frac{B_l^z - \mu_2}{\sigma_2}) - \phi(\frac{A_l^z - \mu_2}{\sigma_2}) \\ - \phi(\frac{B_l^z - \mu_1}{\sigma_1}) + 1 \end{aligned} \tag{5}$$

where ϕ is the cumulative distribution function of the standard Gaussian distribution. A_l^z and B_l^z are intersection points of the two Gaussian distributions of z-th weight in l-th layer of θ^o. We define the gradient similarity for each weight based on the OVL value.

4.2 Weight initialization based on gradient similarity (WIG)

We consider the top p% of OVL values (high gradient similarity) in descending order as important weights for each layer in θ^o. Let $\theta_l^o \in \mathbb{R}^{d(l)}$ be a general weight matrix in θ^o of l-th layer with $\mathcal{D}(l)$ dimension while having L layers for θ^o. ($\theta^o = [\theta_1^o, ..., \theta_L^o]$). To initialize the important weights, we consider a set of gradient similarities $\Phi_l \in \mathbb{R}^{d(l)}$, $\Phi_l = [\Phi_l^1, ..., \Phi_l^{d(l)}]$ by Equation 5 as the similarity value for weight initialization in the l-th layer. Following our strategy, we compute a mask in l-th layer $\mathbf{M}_l \in \mathbb{R}^{d(l)}$ with top p% for the highest values of important weights based on the following element-wise binarization function Top_p: $\mathbb{R}^{d(l)} \to \mathbb{R}^{d(l)}$:

$$\begin{aligned} &\forall z \in \{1, \cdots, d(l)\}, \\ &\mathrm{Top}_p(\Phi_l^z)_z = \begin{cases} 1, & \Phi_l^z \geq \lambda_l^p \\ 0, & \text{Otherwise} \end{cases} \end{aligned} \tag{6}$$

Algorithm 1 Weight Initialization based on Gradient similarity (WIG)

Input: Original model θ^o, Zero vector M, Retain set $\mathcal{D}^R$, Forget set $\mathcal{D}^F$ and the number of layers in the original model L, The initialization rate $p\%$
Compute gradient distributions of $\mathcal{D}$ and $\mathcal{D}^F$ per weight
for l = 1, 2, .., L **do**
  Compute similarity Φ_l^z for each weight in l-th layer by Equation 5
  Calculate $\mathbf{M}_l = \text{Top}_p(\Phi_l)$ by Equation 6
  $\theta_l^W \leftarrow \mathbf{M}_l \odot N(0, \frac{2}{n_{l+1}}) + (1 - \mathbf{M}_l) \odot \theta_l^o$
end for
Fine-tune θ^W using $\mathcal{D}^R$
Output: Unlearned model θ^u with WIG

where λ_l^p is the threshold calculated as the top $p\%$ value with respect to Φ_l in the l-th layer. Then, we apply the calculated mask $\mathbf{M}_l$ to the l-th layer of θ_l^o, and we initialize the important weights using a Kaiming normal distribution [16]. We define the initialized model with the selected important weights from the θ^o as θ^W. Then, we can calculate the layer-wise initialization as follows:

$$\theta_l^W \leftarrow \mathbf{M}_l \odot N(0, \frac{2}{n_{l+1}}) + (1 - \mathbf{M}_l) \odot \theta_l^o \tag{7}$$

where n_{l+1} is the number of nodes in the next layer and $\odot$ is an element-wise product. Therefore, we can effectively remove weights that are potentially harmful to catastrophic forgetting during later fine-tuning. After initializing with WIG, we fine-tune θ^W using $\mathcal{D}^R$ to obtain the unlearned model θ^u.

4.3 Pre-computed Gaussian distribution

We explicitly model the gradients as a Gaussian distribution P parameterized by their mean and variance. Recalculating $P_{\mathcal{D}^R}$ for every unlearning request is computationally expensive, as $\mathcal{D}^R$ is dynamically determined after specific $\mathcal{D}^F$ is selected ($\mathcal{D}^R = \mathcal{D} \setminus \mathcal{D}^F$). Therefore, we take a similar approach from Foster et al. [10] by replacing $P_{\mathcal{D}^R}$ with a pre-computed $P_{\mathcal{D}}$. $P_{\mathcal{D}}$ can be viewed as a weighted mixture of $P_{\mathcal{D}^F}$ and $P_{\mathcal{D}^R}$: $P_{\mathcal{D}} = (1-\alpha)P_{\mathcal{D}^R} + \alpha P_{\mathcal{D}^F}$. During gradient similarity calculation, the component contributed by P_F acts as a self-similarity term that is constant across all parameters, since $OVL(P_{\mathcal{D}^F}, P_{\mathcal{D}^F}) = 1$. Consequently, the inclusion of $\mathcal{D}^F$ effectively introduces a uniform constant bias to every parameter's score. Since WIG employs a Top-K selection, this uniform bias does not alter the relative selection order. This ensures the method remains robust to the size of $\mathcal{D}^F$. We empirically validate this design choice in our ablation study (Section 5.3), demonstrating that WIG achieves unlearning performance virtually identical to that of the $\mathcal{D}^R$ distribution. The experiments of WIG throughout our paper are conducted using the Gaussian distribution derived from $\mathcal{D}$ and $\mathcal{D}^F$. The overall process of WIG is illustrated in Algorithm 1.

4.4 The effect of partial initialization

The difference in loss values between θ^W and the θ^{Gold} is smaller than the difference between the loss values of the θ^o and the θ^{Gold} when $\mathcal{D}^R$ and $\mathcal{D}^F$ are drawn from different distributions.

$$\theta^o = \arg\min_\theta \left(\frac{1}{N}\left\{\sum_j^{|\mathcal{D}^R|} \mathcal{H}(y_j, \hat{y}_j|\theta) + \sum_k^{|\mathcal{D}^F|} \mathcal{H}(y_k, \hat{y}_k|\theta)\right\}\right) \tag{8}$$

$$\begin{aligned}
&\frac{1}{N}\left\{\sum_j^{|\mathcal{D}^R|} \mathcal{H}(y_j, \hat{y}_j|\theta^o) + \sum_k^{|\mathcal{D}^F|} \mathcal{H}(y_k, \hat{y}_k|\theta^o)\right\} \\
&\leq \frac{1}{N}\left\{\sum_j^{|\mathcal{D}^R|} \mathcal{H}(y_j, \hat{y}_j|\theta^W) + \sum_k^{|\mathcal{D}^F|} \mathcal{H}(y_k, \hat{y}_k|\theta^o)\right\} \\
&\leq \frac{1}{N}\sum_j^{|\mathcal{D}^R|} \mathcal{H}(y_j, \hat{y}_j|\theta^W) + \frac{1}{N}\sum_k^{|\mathcal{D}^F|} \mathcal{H}(y_k, \hat{y}_k|\theta^W) \qquad (9) \\
&\frac{1}{N}\sum_k^{|\mathcal{D}^F|}\{\mathcal{H}(y_k, \hat{y}_k|\theta^{Gold}) - \mathcal{H}(y_k, \hat{y}_k|\theta^W)\} \\
&\leq \frac{1}{N}\sum_k^{|\mathcal{D}^F|}\{\mathcal{H}(y_k, \hat{y}_k|\theta^{Gold}) - \mathcal{H}(y_k, \hat{y}_k|\theta^o)\}
\end{aligned}$$

where $\frac{1}{N}\sum_k^{|\mathcal{D}^F|}\{\mathcal{H}(y_k, \hat{y}_k|\theta^{Gold})\} \geq \frac{1}{N}\sum_k^{|\mathcal{D}^F|}\{\mathcal{H}(y_k, \hat{y}_k|\theta^W)\}$. This is because θ^{Gold} is found by $\mathcal{D}^R$, and θ^W is partially initialized from the flat basin optimized by $\mathcal{D}$ containing $\mathcal{D}^F$. When fine-tuning with $\mathcal{D}^R$, a partially initialized model starts from a point on the loss surface closer to the θ^{Gold}, which would possibly lead to better unlearning by reducing the distance from the θ^{Gold}. However, partial initialization degrades $\mathcal{D}^R$ and $\mathcal{D}^{Te}$ accuracy, necessitating additional $\mathcal{D}^R$ fine-tuning, which may inadvertently restore $\mathcal{D}^F$ information due to gradient similarities. This highlights the need for a partial initialization strategy that ensures catastrophic forgetting of $\mathcal{D}^F$, which WIG effectively achieves. Further analysis is provided in Section 5.3.

5 Experiments

5.1 Experimental setup

Datasets and models. We conducted a comprehensive evaluation across various model architectures (ResNet-18 [15], ALL-CNN [36], VGG-16 [35]) and datasets (CIFAR-10 [23], CIFAR-100 [23], SVHN [29]), exploring all combinations. Furthermore, as presented later in Section 5.2, we extend our experiments to include ViT [7] on TinyImageNet [25] and ResNet-18 [15] on ImageNet [6]. Each experiment was repeated for three independent trials, except for the ImageNet experiment, which was conducted using a single seed. Unless noted otherwise, our paper will mainly focus on experiments using ResNet-18 [15] on CIFAR-10 [23] dataset, including the ablation study (Section 5.3).

Implementation details. We utilized the official repositories of our baseline algorithms (introduced in Section 3) and conducted experiments on RTX A5000 and 3090 GPU. Following prior research, we selected class 5 for class and selective forgetting in the Kurmanji et al. [24] environment, class 0 for the Jia et al. [19] environment, classes 0 and 1 for the Inter-class confusion test, and a random 10% sample for random forgetting. We employed SCRUB+R for class, random, and selective unlearning, and SCRUB for the Inter-class confusion test [11], U-LiRA, NeurIPS Machine Unlearning Challenge, and the Gaussian Poison. Prioritizing unlearning performance over generalization, we performed a grid search for each task to minimize the average gap while maintaining at least 90% of the gold model's accuracy. If this was not achievable, we selected

Table 1: Random forgetting results (ResNet-18, CIFAR-10). "()" indicates the performance gap against Retrain. Smaller gaps are better for MIA loss, entropy, confidence, and $\mathcal{D}^F$ Acc, whereas higher values are preferred for $\mathcal{D}^R$ and $\mathcal{D}^{Te}$ Acc.

Method	*MIA loss*	*MIA entropy*	*MIA confidence*	$\mathcal{D}^F$ *Acc*	*Avg Gap*	$\mathcal{D}^R$ *Acc*	$\mathcal{D}^{Te}$ *Acc*
Retrain	49.28±0.25	61.62±0.68	37.37±0.36	82.33±0.10	0.00	100.00±0.00	81.79±0.20
Finetune	66.00±0.50(16.7)	91.48±0.85(29.9)	2.07±0.53(35.3)	100.00±0.00(17.7)	24.89	**100.00±0.00**	82.44±0.06
GA	65.35±0.47(16.1)	98.05±0.53(36.4)	0.37±0.21(37.0)	99.98±0.02(17.7)	26.79	**100.00±0.00**	82.28±0.15
NegGrad	66.16±0.35(16.9)	91.84±0.48(30.2)	1.52±0.34(35.8)	100.00±0.00(17.7)	25.15	**100.00±0.00**	82.38±0.14
CFK	66.34±0.25(17.1)	96.53±0.25(34.9)	0.35±0.13(37.0)	100.00±0.00(17.7)	26.67	**100.00±0.00**	82.34±0.10
EUK	52.89±0.44(3.6)	50.62±8.20(11.0)	18.16±2.97(19.2)	84.08±1.07(1.8)	8.89	84.92±0.24	73.59±0.18
Bad-T	59.27±0.61(10.0)	64.14±0.88(2.5)	18.07±1.00(19.3)	98.09±0.12(15.8)	11.89	99.78±0.01	75.82±0.14
SCRUB+R	55.64±0.64(6.4)	69.04±3.13(7.4)	19.56±1.53(17.8)	91.87±0.95(9.5)	10.28	98.89±1.24	78.43±0.62
SSD	61.33±3.69(12.0)	86.17±10.45(24.6)	7.71±6.21(29.7)	95.88±4.88(13.5)	19.95	96.51±4.03	79.10±2.84
SalUn	**49.02±0.63**(0.3)	54.18±2.21(7.4)	43.12±2.45(5.8)	**83.20±0.36**(0.9)	3.58	99.91±0.01	81.95±0.31
SHs	52.79±0.59(3.5)	69.02±1.32(7.4)	**35.15±1.96**(2.2)	75.03±0.19(7.3)	5.11	99.73±0.15	79.35±0.16
WIG	52.06±0.37(2.8)	**60.18±0.28**(1.4)	34.34±0.25(3.0)	87.88±0.27(5.5)	**3.20**	**100.00±0.00**	**83.64±0.31**

Table 2: Class forgetting results (ResNet-18, CIFAR-10). "()" indicates the performance gap against Retrain. Smaller gaps are better for MIA loss, entropy, confidence, and $\mathcal{D}^F$ Acc, whereas higher values are preferred for $\mathcal{D}^R$ and $\mathcal{D}^{Te}$ Acc.

Method	*MIA loss*	*MIA entropy*	*MIA confidence*	D^F *Acc*	*Avg Gap*	D^R *Acc*	D^{Te} *Acc*
Retrain	50.03±1.89	44.62±1.40	100.00±0.00	0.00±0.00	0.00	100.00±0.00	84.89±0.07
Finetune	70.17±1.30(20.1)	8.37±0.74(36.2)	87.82±0.81(12.2)	71.16±1.26(71.2)	34.93	**100.00±0.00**	84.75±0.06
GA	59.67±0.48(9.6)	21.40±0.37(23.2)	96.46±0.21(3.5)	6.08±0.31(6.1)	10.62	86.65±0.04	73.74±0.35
NegGrad	64.50±0.24(14.5)	16.35±0.16(28.3)	98.59±0.14(1.4)	5.21±0.19(5.2)	12.34	99.19±0.20	81.84±0.25
CFK	70.57±1.19(20.5)	9.58±0.66(35.0)	86.81±0.76(13.2)	74.29±0.37(74.3)	35.77	**100.00±0.00**	84.62±0.06
EUK	**50.57±2.84**(0.5)	31.91±12.92(12.7)	**100.00±0.00**(0.0)	**0.00±0.00**(0.0)	3.31	87.90±0.28	77.04±0.16
Bad-T	69.13±1.39(19.1)	3.73±0.44(40.9)	97.70±0.16(2.3)	29.52±0.53(29.5)	22.95	99.97±0.01	83.73±0.01
SCRUB+R	54.80±0.99(4.8)	**46.40±0.95**(1.8)	**100.00±0.00**(0.0)	**0.00±0.00**(0.0)	1.64	99.99±0.00	84.66±0.08
SSD	53.80±1.50(3.8)	36.36±13.97(8.3)	**100.00±0.00**(0.0)	**0.00±0.00**(0.0)	3.01	77.90±6.22	67.39±4.61
SalUn	56.33±1.78(6.3)	35.23±1.75(9.4)	**100.00±0.00**(0.0)	**0.00±0.00**(0.0)	3.92	**100.00±0.00**	**86.20±0.15**
SHs	48.60±0.64(1.4)	50.51±1.72(5.9)	100.00±0.00(0.0)	**0.00±0.00**(0.0)	1.83	**100.00±0.00**	83.54±0.30
WIG	51.83±1.84(1.8)	42.33±0.94(2.3)	**100.00±0.00**(0.0)	**0.00±0.00**(0.0)	**1.02**	**100.00±0.00**	80.83±0.83

the parameters that yielded the highest test accuracy. The code is available at https://github.com/Ldoun/WIG. Also, note that we used a batch size of 32 for the forget loader of WIG when WIG collected the gradients of the forget set before initialization. This ensures the creation of a Gaussian distribution for the unlearning scenario with the smallest forget sample size (selective forgetting).

5.2 Experiment results

Comparison between different algorithms. For this comparison, we followed the experimental settings outlined in [24]. To maintain consistency with prior research, we first pre-trained the model on CIFAR-100 for all CIFAR-10 experiments, and subsequently fine-tuned θ^o on $\mathcal{D}$ for 30 epochs. We compare models unlearned by several methods, including Fine-tuning, Bad-T, CFK, EUK, SCRUB+R, NegGrad, SalUn, SSD, and WIG across all unlearning scenarios. Comparisons with SHs were conducted exclusively on the ResNet-18 on CIFAR-10 experiments. Tables 1 through 2 present the performance of these unlearning algorithms in the random and class forgetting scenarios on CIFAR-10. The results show that state-of-the-art methods and others do not consistently show small gaps across different model properties. Conversely, WIG consistently achieves small gaps across all tables and metrics, demonstrating the best indistinguishability. Figure 3 further illustrates WIG's robust performance across diverse models, datasets, and unlearning scenarios.

Sparsity aware unlearning. Following the experimental setup and using approximate unlearning algorithms implemented in Jia et al. [19], we conducted experiments on a sparsity-aware unlearning scenario, which follows the procedure 'prune first, then unlearn.' The sparse model was trained using one-shot magnitude-based pruning with rewind [19, 27] to achieve 95% sparsity, and the approximate unlearning was performed on the sparse model. Tables 3 and 4 demonstrate the performance comparison between WIG and other approximate unlearning algorithms in the random and class forgetting scenarios. WIG shows outstanding performance compared to others, even when the sparsity of the model reduces the gap between the approximate and exact unlearning. Figure 4

Table 3: Sparsity-aware random forgetting results (ResNet-18, CIFAR-10). "()" indicates the performance gap against Retrain. Smaller gaps are better for MIA loss, entropy, confidence, and $\mathcal{D}^F$ Acc, whereas higher values are preferred for $\mathcal{D}^R$ and $\mathcal{D}^{Te}$ Acc.

Method	*MIA loss*	*MIA entropy*	*MIA confidence*	$\mathcal{D}^F$ *Acc*	*Avg Gap*	$\mathcal{D}^R$ *Acc*	$\mathcal{D}^{Te}$ *Acc*
Retrain	49.82±0.48	84.08±1.05	13.72±0.66	93.77±0.61	0.00	99.98±0.01	93.40±0.14
Finetune	51.16±0.33(1.3)	74.56±0.70(9.5)	18.09±2.41(4.4)	90.43±1.62(3.3)	4.64	93.37±1.86	88.28±1.61
GA	52.70±1.63(2.9)	89.32±4.25(5.2)	**12.68±8.62**(1.0)	90.73±7.48(3.0)	3.05	91.12±7.31	85.79±6.79
FF	52.86±1.25(3.0)	89.95±0.25(5.9)	11.04±0.75(2.7)	90.07±0.71(3.7)	3.82	89.95±0.55	85.22±0.28
IU	54.33±0.09(4.5)	96.12±0.27(12.0)	2.02±0.17(11.7)	99.27±0.11(5.5)	8.44	**99.36±0.05**	**93.90±0.33**
WIG	**51.15±0.46**(1.3)	**82.92±0.38**(1.2)	11.19±0.42(2.5)	**94.89±0.52**(1.1)	**1.54**	98.09±0.24	92.72±0.21

Table 4: Sparsity-aware class forgetting results (ResNet-18, CIFAR-10). "()" indicates the performance gap against Retrain. Smaller gaps are better for MIA loss, entropy, confidence, and $\mathcal{D}^F$ Acc, whereas higher values are preferred for $\mathcal{D}^R$ and $\mathcal{D}^{Te}$ Acc.

Method	*MIA loss*	*MIA entropy*	*MIA confidence*	D^F *Acc*	*Avg Gap*	D^R *Acc*	D^{Te} *Acc*
Retrain	52.67±1.82	32.54±0.27	100.00±0.00	0.00±0.00	0.00	99.98±0.01	93.77±0.26
Finetune	53.13±0.46(0.5)	50.42±3.62(17.9)	**100.00±0.00**(0.0)	**0.00±0.00**(0.0)	4.59	85.92±2.49	83.11±2.24
GA	52.27±2.19(0.4)	**28.99±0.15**(3.6)	93.84±0.76(6.2)	11.15±1.09(11.2)	5.31	96.43±0.46	90.42±0.30
FF	54.27±0.66(1.6)	97.95±0.37(65.4)	0.93±0.31(99.1)	99.73±0.17(99.7)	66.45	**99.65±0.25**	**94.39±0.26**
IU	54.93±1.27(2.3)	17.28±2.38(15.3)	96.04±2.81(4.0)	10.62±6.15(10.6)	8.02	97.49±0.99	91.61±1.16
WIG	**52.50±1.68**(0.2)	26.48±1.65(6.1)	**100.00±0.00**(0.0)	**0.00±0.00**(0.0)	**1.56**	98.07±0.27	93.41±0.29

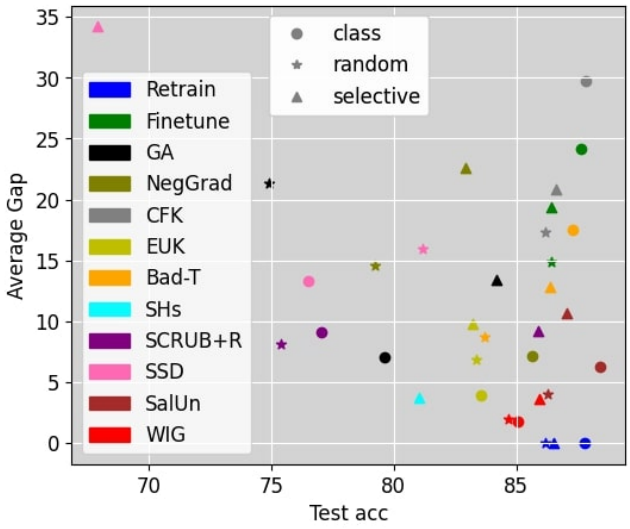


Figure 3: Average gap and test accuracy averaged across models and datasets are presented. Color denotes the method, and shape distinguishes unlearning scenarios. The blue marker represents ideal performance via retraining. Consequently, unlearning results that cluster closer to this blue marker are considered better.

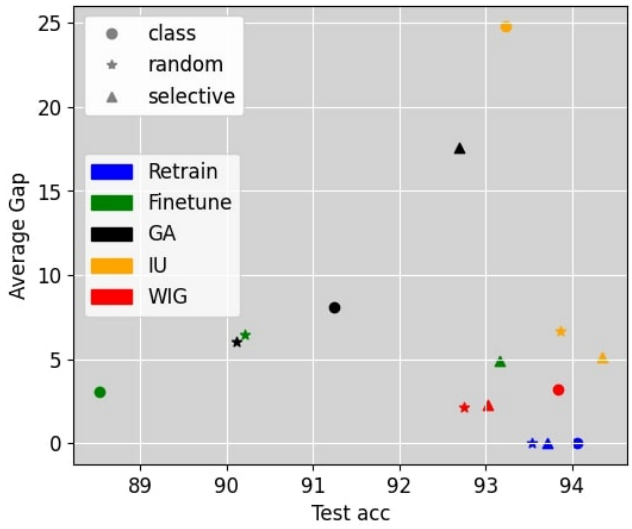


Figure 4: Under sparsity-aware unlearning settings, average gap and test accuracy averaged across models and datasets are presented. Color denotes the method, and shape distinguishes unlearning scenarios. The blue marker represents ideal performance via retraining. Consequently, unlearning results that cluster closer to this blue marker are considered better.

further illustrates WIG's robust performance across diverse models, datasets, and unlearning scenarios.

Inter-class confusion test. The Inter-class confusion test involves adversarially manipulating labels between two selected classes to measure the memorization of $\mathcal{D}^F$ and its influence on unseen data. We report the results for ResNet-18 and VGG-16 in Tables 5 and 6, comparing the performance of various unlearning baselines.

Table 5: Inter-class confusion test results (ResNet-18, CIFAR-10). Lower values are better for Targeted Error, whereas higher values are preferred for $\mathcal{D}^{Te}$ Acc.

Method	*Targeted Error* (↓) $\mathcal{D}^F$	$\mathcal{D}^{Te}$	$\mathcal{D}^{Te}$ *Acc* (↑)
Retrain	51.67±3.30	30.67±1.89	81.63±0.19
Finetune	3470.67±105.99	297.33±22.95	78.61±0.34
GA	836.33±31.48	112.67±7.13	64.35±0.40
NegGrad	675.33±20.82	69.00±5.72	74.09±0.69
CFK	3772.33±44.06	346.33±19.60	77.94±0.28
EUK	276.33±25.09	89.67±3.30	74.79±0.01
Bad-T	382.00±31.84	63.00±3.27	77.81±0.16
SCRUB	139.67±12.23	31.33±1.70	81.19±0.15
SSD	3357.00±736.19	485.67±157.97	73.22±1.57
SalUn	335.33±40.50	45.67±3.40	**83.29±0.13**
SHs	87.33±11.44	33.33±5.44	80.16±0.14
WIG	**83.00±2.16**	**28.00±0.82**	83.20±0.16

Table 6: Inter-class confusion test results (VGG-16, CIFAR-10). Lower values are better for Targeted Error, whereas higher values are preferred for $\mathcal{D}^{Te}$ Acc.

Method	*Targeted Error* (↓) Forget set	Test set	*Test Accuracy* (↑)
Retrain	42.33±4.50	23.33±1.25	82.38±0.19
Finetune	3179.33±354.99	278.33±51.53	81.40±0.44
GA	1269.00±139.29	180.00±37.69	67.01±0.41
NegGrad	1139.33±276.31	65.00±24.06	78.19±0.47
CFK	3781.00±99.40	525.00±26.87	77.58±0.34
EUK	970.00±248.95	165.33±26.91	79.39±0.45
Bad-T	1323.67±123.45	70.67±7.59	81.54±0.49
SCRUB	493.00±159.10	85.00±35.39	82.30±1.04
SSD	3586.67±559.22	855.33±215.31	71.56±1.39
SalUn	335.33±42.62	45.00±0.82	**84.46±0.14**
WIG	**52.67±15.63**	**20.67±5.25**	78.58±0.98

Table 7: Approximate unlearning evaluated using U-LiRA. Algorithms that achieve an Attack Acc close to 50% while maintaining higher $\mathcal{D}^{Te}$ Acc are considered better.

	Finetune	SVD-based	SalUn	WIG
Attack Acc	75.46	75.10	81.64	59.56
$\mathcal{D}^{Te}$ Acc	81.91	85.09	86.71	81.50

Existing algorithms were unable to fully remove the influence of $\mathcal{D}^F$ or weren't able to maintain accuracy on $\mathcal{D}^{Te}$ while removing the influence of $\mathcal{D}^F$. In contrast, WIG significantly outperforms other algorithms by showing minimal targeted errors on $\mathcal{D}^F$ and $\mathcal{D}^{Te}$ while maintaining high accuracy on $\mathcal{D}^{Te}$. Figure 5 further illustrates WIG's robust performance across diverse models and datasets.

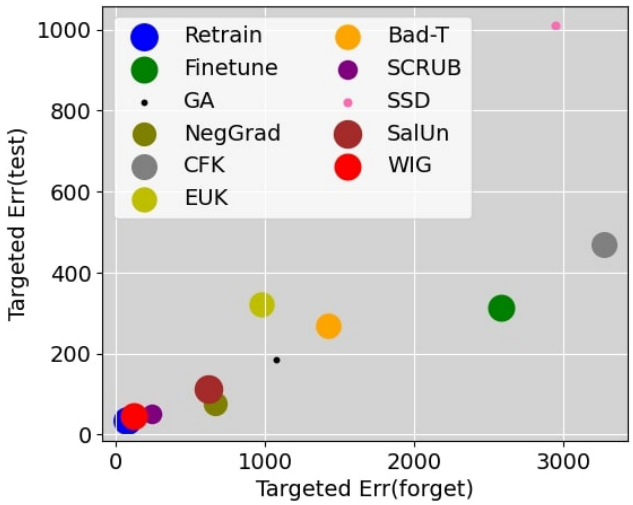


Figure 5: Averaged results of the Inter-class confusion test. Marker color indicates the method, while marker size corresponds to the average test accuracy. Performance is considered better for larger markers positioned closer to the bottom-left corner.

Table 8: Approximate unlearning evaluated on the Machine Unlearning Challenge. Higher scores indicate better performance. We report the average and standard deviation over 10 runs.

	Finetune	Bad-T	NegGrad	SCRUB	SalUn	WIG
Avg	0.061	0.0	0.045	0.060	0.063	**0.087**
Std	0.002	0.0	0.003	0.003	0.003	0.003

U-LiRA. We followed the experimental setup and baselines of Kodge et al. [21] including their proposed SVD-based algorithm. Table 7 demonstrates that our method achieves superior unlearning performance by exhibiting attack accuracy closest to random guessing while maintaining high model utility.

NeurIPS 2023 - Machine unlearning challenge. We utilized the submission system provided by Kaggle to evaluate the unlearning algorithms. In this evaluation, we compare Finetune, Bad-T, SCRUB, NegGrad, SalUn, and WIG. Table 8 presents the average and standard deviation of the score for each algorithm. The table shows that WIG achieves the best score in this evaluation. Additionally, a score of 0.0 can be found when the attack perfectly separates the θ^u and θ^{Gold} distributions, which implies a failure for unlearning. This evaluation employs a confidence-based threshold attack. The ranking presented in Table 8 aligns with the MIA confidence ranking observed in Figure 1a. This empirical evidence supports the hypothesis that biased unlearning can overfit on specific output properties.

Table 9: Results of approximate unlearning evaluated using Gaussian poison. TPR close to 1% is considered better

	Finetune	CFK	EUK	SCRUB	WIG
TPR at FPR=1%	2.93	3.24	1.38	1.02	1.02
$\mathcal{D}^{Te}$ Acc	84.69	84.70	71.51	85.60	81.37

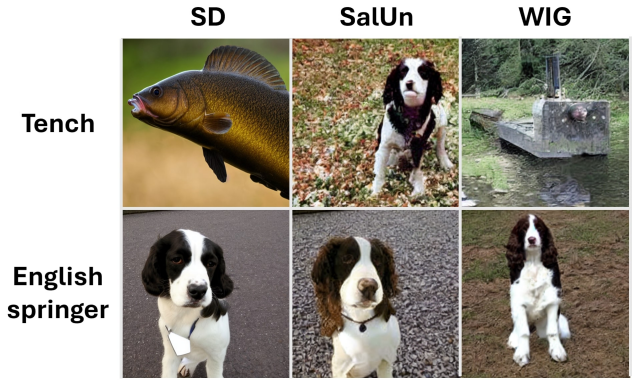


Figure 6: Generation results from Stable Diffusion (SD) and unlearning results from SalUn and WIG when the class "Tench" is unlearned. Rows correspond to the prompts used for image generation, and columns correspond to the generation model.

Gaussian poison. The results in Table 9 demonstrate that WIG and SCRUB achieve successful unlearning in contrast to other baseline methods. These findings refute the limitations of approximate unlearning against Gaussian poisoning as suggested by Pawelczyk et al. [31]. We attribute the success of SCRUB in our experiments to a broader hyperparameter search than that employed in the original work.

Stable Diffusion class unlearning. Following the experimental setup of [9], we conducted class unlearning experiments on the Stable Diffusion (SD) model [32] using the ImageNet dataset [6], comparing our results with SalUn. In this setup, WIG successfully unlearned the Tench prompt by generating images that do not depict "Tench" for any given "Tench" prompt. Examples of the generated results can be found in Figure 6. Additionally, an important observation is that SalUn internally redirects the "Tench" prompt to generate images of an "English Springer". This behavior prevents it from functioning like a fully retrained model that excludes "Tench" during training, thus leaving traces of the unlearning (generating "English Springer" images for the "Tench" prompt). In contrast, since WIG approximates retraining, it behaves like an actual retrained model, distinguishing it from SalUn.

Extending to diverse architectures and datasets. This section evaluates the performance of WIG using Vision Transformer [7] (ViT) on TinyImageNet [25] and ResNet-18 on ImageNet [6]. Table 10 demonstrates WIG's effectiveness in a random forgetting scenario with ViT on TinyImageNet, showcasing the smallest average gap.

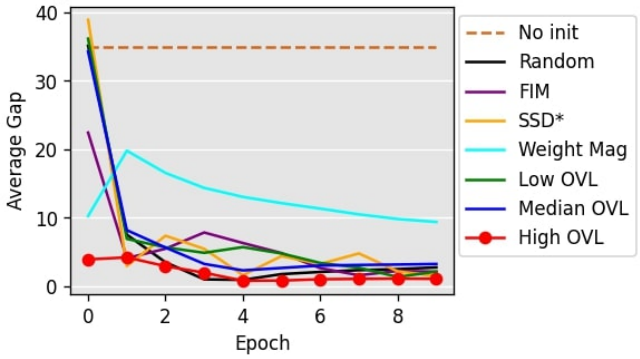


Figure 7: Fine-tuning results from various partial initialization strategies. The plot demonstrates the changes in the average gap as training progresses. An upward trend in the gap can be interpreted as a resurgence of the $\mathcal{D}^F$'s influence. Additionally, "No Init" represents the average gap result of fine-tuning.

Similarly, Table 11 shows WIG's effectiveness in a random forgetting scenario with ResNet-18 on ImageNet, also exhibiting the smallest average gap. Notably, as shown in Table 11, WIG achieves higher $\mathcal{D}^R$ accuracy and lower $\mathcal{D}^F$ accuracy compared to the θ^{Gold}, despite being initially trained on $\mathcal{D}^F$. This indicates that WIG successfully recovered information about $\mathcal{D}^R$ while minimizing the influence of $\mathcal{D}^F$, even though both datasets are drawn from the same distribution in this random forgetting scenario.

5.3 Ablation study

Weight selection. We demonstrate the effectiveness of WIG across various design choices by comparing its performance with different weight selection strategies in a class forgetting scenario. These include random weight selection (Random), weight selection based on high Fisher information on $\mathcal{D}^F$ (FIM), large weight magnitude (Weight Mag), Low OVL, Median OVL, High OVL, and weights with large fisher($\mathcal{D}^F$) − fisher($\mathcal{D}^R$) values (SSD*), which is employed by SSD and DeepClean. Note that High OVL corresponds to WIG, Median OVL selects weights with a central p% OVL value, and Low OVL selects weights with the lowest p% OVL value. Figure 7 illustrates that all weight selection strategies, except High OVL, result in an unstable average gap exhibiting an upward trend during training. This trend suggests that the influence of $\mathcal{D}^F$ can be indirectly magnified by the gradients of $\mathcal{D}^R$. In contrast, High OVL maintains a stable, minimal gap, underscoring its importance for maximizing catastrophic forgetting. Furthermore, as discussed in Section 4.4, all partial initialization strategies outperformed No Init (equivalent to fine-tuning). In this section, we focus primarily on the class forgetting scenario, where $\mathcal{D}^F$ is drawn from a different distribution than $\mathcal{D}^R$. Unlike the random forgetting scenario, this separation minimizes the risk of $\mathcal{D}^F$ information recovery driven by similarity with $\mathcal{D}^R$. Consequently, the upward trend can also be found for WIG in random and selective forgetting scenarios.

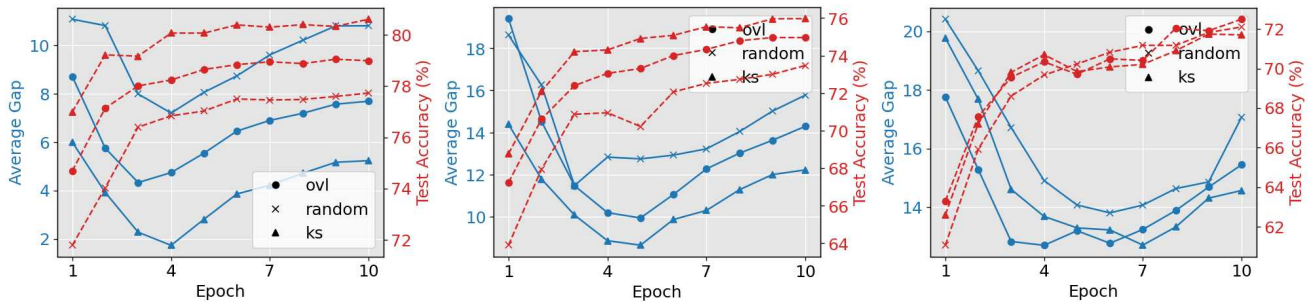
Validation of similarity metric. While our work operates under a Gaussian assumption for gradients, we acknowledge this is an

Table 10: Random forgetting results (ViT, TinyImageNet). "()" indicates the performance gap against Retrain. Smaller gaps are better for MIA loss, entropy, confidence, and $\mathcal{D}^F$ Acc, whereas higher values are preferred for $\mathcal{D}^R$ and $\mathcal{D}^{Te}$ Acc.

Method	*MIA loss*	*MIA entropy*	*MIA confidence*	$\mathcal{D}^F$ *Acc*	*Avg Gap*	$\mathcal{D}^R$ *Acc*	$\mathcal{D}^{Te}$ *Acc*
Retrain	49.91±0.32	76.15±0.65	22.47±0.50	89.35±0.29	0.00	99.61±0.02	89.21±0.05
Finetune	58.18±0.13(8.3)	92.84±0.40(16.7)	7.02±0.30(15.6)	99.23±0.08(9.9)	12.63	**99.97±0.00**	**88.51±0.19**
SalUn	**55.61±0.33(5.7)**	87.18±0.52(11.0)	11.75±0.33(10.7)	97.47±0.00(8.1)	8.88	99.93±0.00	88.31±0.07
WIG	56.22±0.33(6.3)	**85.65±0.62(9.5)**	**14.17±0.34(8.3)**	**96.52±0.28(7.2)**	**7.83**	**99.97±0.00**	86.30±0.07

Table 11: Random forgetting results (ResNet-18, ImageNet). "()" indicates the performance gap against Retrain. Smaller gaps are better for MIA loss, entropy, confidence, and $\mathcal{D}^F$ Acc, whereas higher values are preferred for $\mathcal{D}^R$ and $\mathcal{D}^{Te}$ Acc.

Method	*MIA loss*	*MIA entropy*	*MIA confidence*	$\mathcal{D}^F$ *Acc*	*Avg Gap*	$\mathcal{D}^R$ *Acc*	$\mathcal{D}^{Te}$ *Acc*
Retrain	51.42	45.57	34.72	67.94	0.00	75.50	65.20
Finetune	54.70(3.28)	62.58(17.01)	32.04(2.68)	69.52(1.58)	6.14	**90.12**	60.40
SalUn	54.60(3.18)	**56.97(11.4)**	30.84(3.88)	71.09(3.15)	5.43	80.18	**62.12**
WIG	**53.47(2.05)**	62.40(16.83)	**34.38(0.34)**	**66.71(1.23)**	**5.11**	77.56	59.94



(a) 30% partial initialization on CIFAR-5 **(b) 40% partial initialization on CIFAR-5** **(c) 50% partial initialization on CIFAR-5**

Figure 8: Comparison of OVL, Random, and KS weight selection strategies on the CIFAR-5 dataset. Each plot shows the Test Accuracy and Average Gap over 10 epochs for a different partial initialization percentage: (a) 30%, (b) 40%, and (c) 50%.

idealization. To validate our method's robustness, we compare it against a non-parametric alternative, the Kolmogorov-Smirnov (K-S) statistic, which makes no such distributional assumptions. A significant drawback of the K-S statistic is its computational cost. Its memory requirement scales with both the number of parameters and the number of gradient samples (e.g., $O(D \cdot N)$). In contrast, OVL scales only with the number of parameters ($O(D)$). Due to these severe memory constraints, we performed this experiment on a subset of the CIFAR-10 dataset to avoid out-of-memory (OOM) errors. This subset contains only 5 selected classes from CIFAR-10 (referred to as CIFAR-5). The results of this comparison are shown in Figure 8, which compares OVL, Random, and K-S at three different initialization rates under the CIFAR-5 random forgetting scenario. All results report the average of 3 runs. The plots consistently demonstrate that K-S outperforms the other methods, while OVL achieves the second-best results, and the random baseline performs the worst. This result is significant because OVL is far more memory-efficient and faster than the non-parametric alternative, which requires processing all samples. This confirms that OVL serves as an effective and efficient proxy for a more complex non-parametric measurement.

Validation of the pre-computed $\mathcal{D}$ gradient distribution. As mentioned in Section 4.3, we use a pre-computed Gaussian distribution (derived from the $\mathcal{D}$). To validate this approach, we compare the performance of WIG when using each distribution. Figure 9 plots the performance of WIG using $\mathcal{D}$ distribution against the performance of WIG using the true $\mathcal{D}^R$ distribution, across different $p\%$ values. The results, averaged over 10 runs, show that the performance using $\mathcal{D}$ is nearly identical to the performance using the true $\mathcal{D}^R$. This confirms that the pre-computed $\mathcal{D}$ distribution serves as an effective replacement for the true $\mathcal{D}^R$ distribution.

Multiple unlearning requests. While we acknowledge the concern regarding "overfitting mitigation", our ablation on multiple unlearning requests (Table 12) empirically refutes this hypothesis. In this experiment, we unlearn a single class via three sequential selective forgetting requests. If WIG relied on overfitting, the accumulation of multiple unlearning requests would cause rapid degradation of

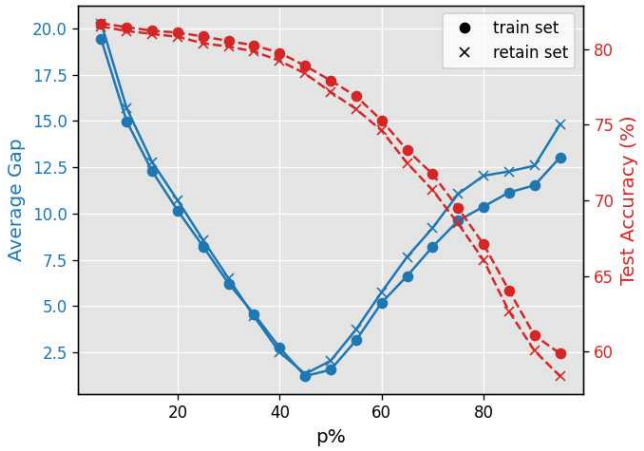


Figure 9: Comparison of the pre-computed gradient distribution ($\mathcal{D}$) and the true gradient distribution ($\mathcal{D}^R$) across different $p\%$ values.

Table 12: Approximate unlearning evaluated under multiple selective forgetting requests. Lower Average Gap and higher $\mathcal{D}^R$ and $\mathcal{D}^{Te}$ accuracies indicate better performance.

Method	Average Gap	$\mathcal{D}^R$ Acc	$\mathcal{D}^{Te}$ Acc
Retrain	0.00	100.00	85.05
Finetune	31.87	100.00	85.67
GA	10.75	89.27	74.98
CFK	30.24	100.00	84.90
EUK	4.87	92.83	79.14
Bad-T	14.61	100.00	84.79
SalUn	11.71	100.00	86.51
WIG	0.44	99.40	83.57

$\mathcal{D}^R$ and $\mathcal{D}^{Te}$ accuracy or fail to unlearn memorized data from previous requests. Instead, we observe steady performance even when sequentially unlearning subsets of a single class.

Sensitivity to initialization rate ($p\%$) and unlearning epochs. Figure 10 illustrates the model's sensitivity to both the initialization rate ($p\%$) and the number of unlearning epochs under a random-forgetting scenario with a constant learning rate. The presented results are averaged over 3 runs. We observe that test accuracy consistently decreases as $p\%$ increases. In contrast, the average gap first decreases (for $p\%$ between 10-50%) and then increases (for $p\%$ between 70-90%). We observed that initialization rates of 30% and 40% frequently achieved the best performance across different unlearning scenarios and model architectures. Furthermore, more unlearning epochs do not necessarily lead to a better unlearned model. In fact, for initialization rates between 50-90%, we found that additional unlearning epochs harm the unlearning performance. This suggests that an early stopping strategy is feasible which reduces the hyperparameter search space by obviating the need to tune the number of epochs. Additionally, we compared our method (WIG) with a random weight selection baseline. While the best performing model from random weight selection shows to be competitive with WIG, it failed to consistently achieve good performance compared to WIG. In contrast, WIG demonstrates robust performance across different initialization rates.

Computation time. Figure 11 details the time required for both preparatory steps and a single unlearning epoch. For instance, preparatory steps involve calculating the Fisher information matrix for SSD, identifying the weight saliency map for SalUn, and constructing the Gaussian distribution for WIG. WIG demonstrates superior unlearning performance with low computational overhead, requiring a minimal amount of processing per epoch. We report full wall-clock times for each algorithm across scenarios with different unlearn epochs in Appendix Table 14.

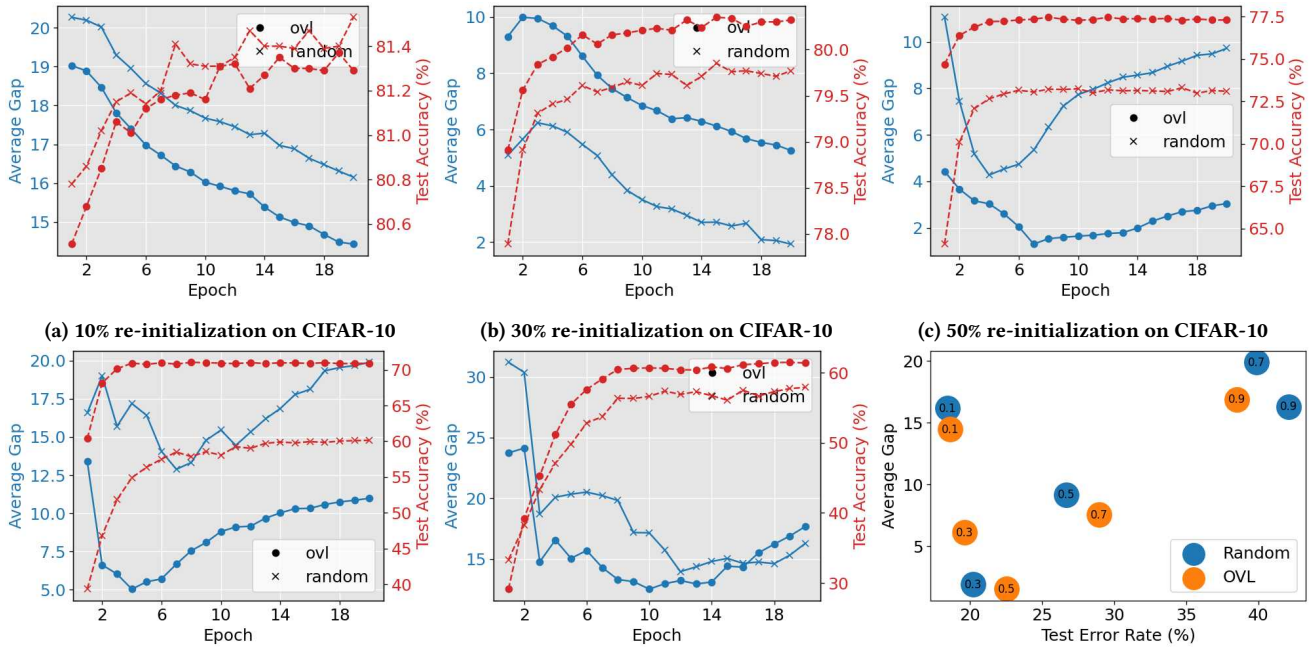
5.4 Discussion and privacy implications

Our evaluations show that WIG achieves a superior and consistent approximation of θ^{Gold} (exact unlearning) than state-of-the-art methods. Here, we discuss the critical privacy implications of why WIG succeeds where biased methods fail.

Output-level unlearning vs. structural unlearning. Most approximate unlearning algorithms, such as SalUn or SCRUB, operate by applying output-level adjustments. These methods are designed to induce unlearning by manipulating specific output properties such as fine-tuning with random labels or using gradient ascent to increase the loss on the forget set. As our results in Figure 1a and Table 1 show, this biased approach fails, as an adaptive attacker can simply query an undefended property (e.g., entropy) to bypass the defense by unlearning. WIG, by contrast, performs a structural unlearning. By identifying and initializing parameters based on gradient similarity, WIG fundamentally alters the model's internal state to more closely approximate the θ^{Gold} model. This mechanism, which approximates the retraining process rather than just its output, is inherently unbiased and more robust.

Evidence of privacy failure in biased unlearning. This distinction is not just theoretical. The Stable Diffusion experiment (Figure 6) provides a clear example of this privacy failure. SalUn's output-level adjustment, does not forget the "Tench" class. It instead creates auditable evidence by redirecting the generation to "English Springer". This trace is a clear privacy failure, as the model's behavior is now a direct artifact of the unlearning process itself. In contrast, WIG, which approximates retraining, correctly behaves like a model that never saw class "Tench".

Evidence of persistent unlearning. Furthermore, our experiments demonstrate that WIG's unlearning is persistent. As shown in Figure 7, all other weight selection strategies (including Random, Weight Mag, FIM, Low OVL, and Median OVL) result in an unstable average gap that trends upward during fine-tuning. This trend indicates that the forgotten information resurfaces as the model is optimized on $\mathcal{D}^R$. By partially initializing weights with high OVL, WIG disrupts the resurfacing of forgotten information driven by weights that share similar gradients between $\mathcal{D}^F$ and $\mathcal{D}^R$. Because these weights encode the shared features, their initialization magnifies catastrophic forgetting and leads to a stable and minimal gap.



(a) 10% re-initialization on CIFAR-10 (b) 30% re-initialization on CIFAR-10 (c) 50% re-initialization on CIFAR-10

(d) 70% re-initialization on CIFAR-10 (e) 90% re-initialization on CIFAR-10 (f) Overall result of best performing model

Figure 10: Sensitivity analysis of initialization rate ($p\%$) and number of unlearning epochs (Random forgetting, CIFAR-10). Subplots (a) through (e) illustrate how $\mathcal{D}^{Te}$ accuracy and average gap evolve over 20 epochs for initialization rates from 10% to 90%. The plots compare partial initialization strategies (OVL, Random) followed by fine-tuning. Subplot (f) reports the best-performing model from each $p\%$ experiment across different number of unlearning epochs.

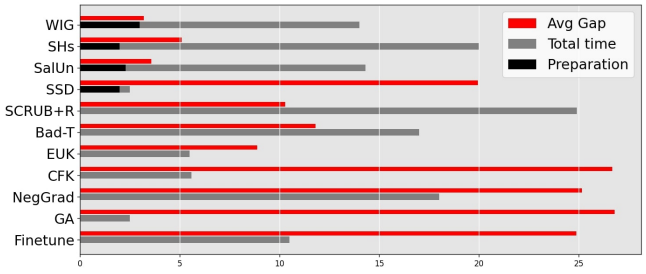


Figure 11: Computation time and average gap measured in the random forgetting scenario. Preparation refers to the time required for the preparatory step before unlearning, while total time includes both preparation and one epoch of unlearning, except for SSD, which requires no additional training.

Limitations and trade-offs. As noted in our ablation study (Figure 10), WIG's strategy of partial initialization based on gradient similarity can inadvertently affect parameters important to the retain set, potentially impacting performance of $\mathcal{D}^R$ and $\mathcal{D}^{Te}$. However, as long as the underlying distributions of $\mathcal{D}^F$ and $\mathcal{D}^R$ overlap or share common features, it is necessary to remove the influence of $\mathcal{D}^F$ on these parameters to achieve robust unlearning. This approach also aligns with the majority of unlearning algorithms that operate through a process of information destruction followed by recovery. Another limitation of our approach is the high memory usage required to store per-weight Gaussian statistics over $\mathcal{D}$ and $\mathcal{D}^F$, which are used to calculate gradient similarity. While this peak cost matches the storage overhead of SCRUB or the memory burden of Bad-T, it is transient. In contrast to baselines that require persistent resource allocation throughout optimization, WIG restricts this heavy computation to the preparatory step.

6 Conclusion

We propose **W**eight **I**nitialization based on **G**radient similarity (WIG) for versatile machine unlearning. Our method maximizes catastrophic forgetting by initializing weights that exhibit high gradient similarity between $\mathcal{D}$ and $\mathcal{D}^F$. Through this approach, WIG magnifies catastrophic forgetting of the target data by initializing weights that encode the shared features between $\mathcal{D}$ and $\mathcal{D}^F$. This process, followed by fine-tuning, structurally approximates the gold standard of exact unlearning. Furthermore, WIG addresses the critical problem of biased forgetting procedure in existing unlearning algorithms, which typically arises from directly modifying specific model properties on $\mathcal{D}^F$. Given the unpredictability of the properties an attacker might exploit, it is crucial to create an unbiased, unlearned model. We demonstrate the superiority of WIG through extensive experiments across various models, datasets, scenarios, and unlearning metrics.

Acknowledgments

We used generative AI-based tools to revise the text, improve flow, and correct any typos, grammatical errors, and awkward phrasing. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2025-23524783)

References

[1] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. 2018. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European conference on computer vision (ECCV)*. 139–154.

[2] Yinzhi Cao and Junfeng Yang. 2015. Towards Making Systems Forget with Machine Unlearning. In *2015 IEEE Symposium on Security and Privacy*. 463–480. https://doi.org/10.1109/SP.2015.35

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

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

[5] R Dennis Cook and Sanford Weisberg. 1982. Residuals and influence in regression. *New York: Chapman and Hall* (1982).

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 248–255. https://doi.org/10.1109/CVPR.2009.5206848

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

[8] Sayna Ebrahimi, Franziska Meier, Roberto Calandra, Trevor Darrell, and Marcus Rohrbach. 2020. Adversarial continual learning. In *European conference on computer vision*. Springer, 386–402.

[9] Chongyu Fan, Jiancheng Liu, Yihua Zhang, Dennis Wei, Eric Wong, and Sijia Liu. 2024. Salun: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. *International Conference on Learning Representations (ICLR)* (2024).

[10] Jack Foster, Stefan Schoepf, and Alexandra Brintrup. 2024. Fast machine unlearning without retraining through selective synaptic dampening. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 12043–12051.

[11] Shashwat Goel, Ameya Prabhu, Amartya Sanyal, Ser-Nam Lim, Philip Torr, and Ponnurangam Kumaraguru. 2022. Towards adversarial evaluations for inexact machine unlearning. *arXiv preprint arXiv:2201.06640* (2022).

[12] Aditya Golatkar, Alessandro Achille, and Stefano Soatto. 2020. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9304–9312.

[13] Aditya Golatkar, Alessandro Achille, and Stefano Soatto. 2020. Forgetting outside the box: Scrubbing deep networks of information accessible from input-output observations. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX 16*. Springer, 383–398.

[14] Jamie Hayes, Ilia Shumailov, Eleni Triantafillou, Amr Khalifa, and Nicolas Papernot. 2025. Inexact unlearning needs more careful evaluations to avoid a false sense of privacy. In *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*. IEEE, 497–519.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. arXiv:1512.03385 [cs.CV]

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*. 1026–1034.

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

[18] Henry F Inman and Edwin L Bradley Jr. 1989. The overlapping coefficient as a measure of agreement between probability distributions and point estimation of the overlap of two normal densities. *Communications in Statistics-theory and Methods* 18, 10 (1989), 3851–3874.

[19] Jinghan Jia, Jiancheng Liu, Parikshit Ram, Yuguang Yao, Gaowen Liu, Yang Liu, Pranay Sharma, and Sijia Liu. 2023. Model sparsity can simplify machine unlearning. *Advances in Neural Information Processing Systems* 36 (2023), 51584–51605.

[20] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences* 114, 13 (2017), 3521–3526.

[21] Sangamesh Kodge, Gobinda Saha, and Kaushik Roy. 2024. Deep unlearning: Fast and efficient gradient-free class forgetting. *Transactions on Machine Learning Research* (2024).

[22] Tatsuya Konishi, Mori Kurokawa, Chihiro Ono, Zixuan Ke, Gyuhak Kim, and Bing Liu. 2023. Parameter-level soft-masking for continual learning. In *International conference on machine learning*. PMLR, 17492–17505.

[23] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).

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

[25] Yann Le and Xuan Yang. 2015. Tiny imagenet visual recognition challenge. *CS 231N* 7, 7 (2015), 3.

[26] David Lopez-Paz and Marc'Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. *Advances in neural information processing systems* 30 (2017).

[27] Xiaolong Ma, Geng Yuan, Xuan Shen, Tianlong Chen, Xuxi Chen, Xiaohan Chen, Ning Liu, Minghai Qin, Sijia Liu, Zhangyang Wang, et al. 2021. Sanity checks for lottery tickets: Does your winning ticket really win the jackpot? *Advances in Neural Information Processing Systems* 34 (2021), 12749–12760.

[28] James Martens. 2020. New insights and perspectives on the natural gradient method. *Journal of Machine Learning Research* 21, 146 (2020), 1–76.

[29] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. 2011. Reading Digits in Natural Images with Unsupervised Feature Learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*. http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf

[30] Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. 2020. What is being transferred in transfer learning? *Advances in neural information processing systems* 33 (2020), 512–523.

[31] Martin Pawelczyk, Jimmy Z Di, Yiwei Lu, Ayush Sekhari, Gautam Kamath, and Seth Neel. 2024. Machine unlearning fails to remove data poisoning attacks. *arXiv preprint arXiv:2406.17216* (2024).

[32] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10684–10695.

[33] Jeffrey Rosen. 2011. The right to be forgotten. *Stan. L. Rev. Online* 64 (2011), 88.

[34] Jialei Shi, Kostis Gourgoulias, John F Buford, Sean J Moran, and Najah Ghalyan. 2024. Deepclean: Machine unlearning on the cheap by resetting privacy sensitive weights using the fisher diagonal. In *European Conference on Computer Vision*. Springer, 1–16.

[35] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[36] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. 2014. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806* (2014).

[37] Eleni Triantafillou, Peter Kairouz, Fabian Pedregosa, Jamie Hayes, Meghdad Kurmanji, Kairan Zhao, Vincent Dumoulin, Julio Jacques Junior, Ioannis Mitliagkas, Jun Wan, et al. 2024. Are we making progress in unlearning? Findings from the first NeurIPS unlearning competition. *arXiv preprint arXiv:2406.09073* (2024).

[38] Jing Wu and Mehrtash Harandi. 2024. Scissorhands: Scrub data influence via connection sensitivity in networks. In *European Conference on Computer Vision*. Springer, 367–384.

[39] Friedemann Zenke, Ben Poole, and Surya Ganguli. 2017. Continual learning through synaptic intelligence. In *International conference on machine learning*. PMLR, 3987–3995.

[40] Shifeng Zhang, Ajian Liu, Jun Wan, Yanyan Liang, Guodong Guo, Sergio Escalera, Hugo Jair Escalante, and Stan Z Li. 2020. Casia-surf: A large-scale multi-modal benchmark for face anti-spoofing. *IEEE Transactions on Biometrics, Behavior, and Identity Science* 2, 2 (2020), 182–193.

[41] Zhen Zhao, Zhizhong Zhang, Xin Tan, Jun Liu, Yanyun Qu, Yuan Xie, and Lizhuang Ma. 2023. Rethinking gradient projection continual learning: Stability/plasticity feature space decoupling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 3718–3727.

A Detailed description of unlearning metrics

Inter-class confusion test. The inter-class confusion test works by unlearning the adversarially manipulated data that negatively affected the original model training. This test is designed to capture the memorization of the forget set and the removal of the forget set's influence on unseen data by measuring the misclassification that occurred in manipulated classes. Intuitively, a successfully unlearned model should not contain any traces of adversarially manipulated data, resulting in lower misclassification rates on the manipulated classes. The inter-class confusion test works in the following steps:

Algorithm 2 Inter-class confusion test

Input: Cl_1, Cl_2: Classes to manipulate
Input: n: Number of single class
1. Select $\frac{2}{n}$ examples each from Cl_1, Cl_2, and swap labels to inter-class.
2. Train the model using the unselected classes and the adversarially manipulated data.
3. Perform unlearning using the manipulated data.
4. Evaluate inter-class confusion test using the actual data.(data before adversarial manipulation)

In this evaluation, Goel et al. [11] used the term targeted error, which is defined as the number of misclassifications that occurred in two selected classes. Specifically, we denote C^D as the confusion matrix for the dataset D, and $C^D_{A,B}$ as the confusion matrix representing the number of examples from class A in D that are misclassified as class B. The targeted errors are then formulated as follows:

$$\text{Targeted Err}(D) = C^D_{A,B} + C^D_{B,A} \tag{10}$$

The targeted errors on the forget set measure the memorization of the forget set. On the other hand, the targeted errors on the test set measure the influence of the forget set on the unseen data.

Machine unlearning challenge. The evaluation mechanism generates an N unlearned model using the participant's proposed algorithm to measure the distribution difference F between the N unlearned and retrained models for the different forget set examples. It is calculated by using the most successful attack performed during M confidence-based threshold attacks given the output of N θ^u and N θ^{Gold} for each example in D^F. If an attack perfectly separates two distributions, the distribution difference F results in zero. The evaluation mechanism of the Machine Unlearning Challenge works as the following algorithm:

Then the final score is calculated using the multiplication of the distribution difference F and the model utility as follows:

$$\text{Score} = F \times \frac{Acc(D^R, \theta^u)}{Acc(D^R, \theta^{Gold})} \times \frac{Acc(D^{Te}, \theta^u)}{Acc(D^{Te}, \theta^{Gold})} \tag{11}$$

$$Acc(D, \theta) = \frac{1}{|D|} \sum_{x,y \in D} [f(x; \theta) = y] \tag{12}$$

Algorithm 3 Distribution difference F

Input: D^F: The forget set
Input: H: Bin function, $H(\delta) = \frac{2}{2^{n(\delta)}}$
Input: nan-max: returns max value while discarding any nan
all-$\delta \leftarrow \{\}$
for s $\in D^F$ **do**
  Calculate FPR and FNR by conducting m attacks on s
  per-attack-$\delta \leftarrow \{\}$
  for i $\in \{0, \ldots, m\}$ **do**
    if FPR[i] = 0 and FNR[i] = 0 **then**
      per-attack-δ.add(inf)
    else if FPR[i] = 0 or FNR[i] = 0 **then**
      pass
    else
      per-attack-$\delta_1 \leftarrow$ log(1 - δ - FPR[i]) - log(FNR[i])
      per-attack-$\delta_2 \leftarrow$ log(1 - δ - FNR[i]) - log(FPR[i])
      per-attack-δ.add(nan-max(per-attack-δ_1, per-attack-δ_2))
    end if
  end for
  all-δ.add(nan-max(per-attack-δ))
end for
$F \leftarrow \frac{1}{|D^F|} \sum_{\delta \in \text{all}-\delta} H(\delta)$
return F

B Additional discussion

B.1 Performance of WIG in Adam-based target model

In this section, we analyze the effect of replacing the SGD-based original model θ^o with an Adam-based counterpart. Table 13 demonstrates that WIG maintains superior performance in the ResNet-18 CIFAR-10 class forgetting scenario even with Adam-trained model. Note that while the original model was trained with Adam, we consistently employed SGD for WIG and all baseline unlearning algorithms. Furthermore, we expanded the hyperparameter search space to include learning rates of (5e-2, 1e-2, 1e-3, 1e-4) as gradient-based methods failed to exhibit robust performance at smaller learning rates.

Table 13: Results of approximate unlearning evaluated using the Adam-based original model (θ^o). Smaller Average Gap and higher $\mathcal{D}^R$ and $\mathcal{D}^{\mathcal{T}\rceil}$ accuracy are considered better.

Method	Average Gap	Retain Acc	Test Acc
Retrain	0.00	97.05	83.72
Finetune	4.68	93.52	80.86
GA	10.72	88.22	76.21
CFK	4.04	99.78	85.82
EUK	2.23	99.48	85.49
Bad-T	23.68	96.84	85.19
SalUn	6.29	87.83	79.89
WIG	0.71	95.13	82.25

B.2 Analysis on wall-clock time

Table 14 presents the wall-clock time for each algorithm across three different unlearning scenarios. Each algorithm was evaluated using distinct hyperparameters optimized for each specific scenario. Note that these measurements were obtained in a different experimental environment than those reported in Figure 11.

C Detailed settings of each experiment

Our experiment adhered to the experimental setups outlined in Kurmanji et al. [24] and Jia et al. [19]. We refer readers to [19, 24] for more detailed information.

Table 14: Full wall-clock time

	Retrain	Finetune	GA	NegGrad	CFK	EUK	Bad-T	SCRUB+R	SSD	SalUn	SHs	WIG
Random forgetting	329.53	90.71	11.11	173.39	38.87	11.77	158.47	102.76	11.22	31.50	305.88	114.35
Selective forgetting	361.17	101.33	0.58	177.48	46.18	13.77	80.12	114.32	11.26	30.78	274.42	67.53
Class forgetting	327.82	90.60	3.43	173.58	38.97	11.78	158.83	59.16	10.15	51.98	290.42	113.95

C.1 Comparison between different algorithms

Models. We followed the same modified version of ResNet-18 [15] and ALL-CNN [36] in Kurmanji et al. [24]. For the VGG-16 [35], we used the version of Jia et al. [19].

Pretraining. For consistency with previous work, we pretrained the model for CIFAR-10[23] experiments using CIFAR-100[23] with a learning rate of 0.1. As for other experiments, we did not adopt any pretraining.

Approximate unlearning. We generally followed the implementation in Kurmanji et al. [24]. However, for Bad-T and EUk, we used the official implementation from Chundawat et al. [4] and Goel et al. [11]. For SalUn and SSD, we implemented the algorithm from the official repository [9, 10].

The hyperparameters for each algorithm were tuned using a grid search. For each algorithm, the unlearning epoch and learning rate were tuned over values in the ranges (1e-2, 1e-3, 1e-4) and (3, 5, 10), respectively. Algorithms with additional hyperparameters were tuned as follows:

- For WIG, reinit_p was tuned over (20, 30, 40, 50).
- For NegGrad, alpha was tuned over (0.9999, 0.5).
- For SCRUB and SCRUB+R, we tuned the retain and forget batch sizes over the set $\{32, 64, 128, 256\}$ and the maximization steps over $\{1, 2, 3, 4, 5\}$. The learning rate and number of epochs were fixed at 0.0005 and 5 respectively. This follows the original methodology which relies on batch size to control convergence. Furthermore, it prioritizes the maximization steps to regulate the unlearning process instead of the total number of unlearning epochs which encompass both minimization and maximization phases.
- For SSD, alpha and lambda were tuned over (1, 2, 3, 5, 10, 25, 50) and (0.1, 0.3, 0.5, 0.7, 1, 2, 3), respectively.

C.2 Sparsity aware unlearning

Models. We used the same version of ResNet-18 [15] and VGG-16 [35] in Jia et al. [19]. We used the version of Kurmanji et al. [24] for the ALL-CNN [36].

Pruning. We trained the sparse and dense models following the pruning setup outlined in Jia et al. [19].

Approximate Unlearning. We primarily followed the implementation provided in Jia et al. [19]. Additionally, we adopted a learning rate scheduler for WIG that linearly increases the target learning rate over the first 5 epochs and then decreases over the remaining 5 epochs. The hyperparameters for each algorithm were tuned using a grid search. For each algorithm, the learning rate was tuned over values in the ranges (1e-1, 1e-2, 1e-3, 1e-4, 1e-5) while the epoch was fixed at 10. Algorithms with additional or different hyperparameters were tuned as follows:

- For WIG, reinit_p was tuned over (20, 30, 40, 50).
- For Fisher forgetting, alpha was tuned over (1e-9, 1e-8, 1e-7).
- For influence function, alpha was tuned over (0.1 0.5 1).

C.3 Kaggle NeurIPS 2023 - Machine unlearning competition

Hyper-parameters. In this competition setting, we tuned the hyperparameters of each method using a grid search. The learning rate was found in a range of (1e-2, 1e-3, 5e-4, 1e-4), and epoch was found using a range of (1, 2, 3, 4, 5). Additionally, we adopted a cosine learning rate scheduler for fine-tuning and WIG. SCRUB used a max-step of 1, and SalUn used 0.5 for the gradient magnitude mask threshold. Also, the initialization value p of WIG is fixed at 30%.

Table 15: The hyper-parameters of the approximate algorithms evaluated in the NeurIPS 2023 Machine Unlearning Challenge [37] are presented.

Method	Epoch	Learning Rate (lr)	Score
Fine-tuning	1	0.001	0.061
Bad-T	1	0.001	0.0
NegGrad	1	0.001	0.045
SCRUB	1	0.0005	0.060
SalUn	4	0.001	0.063
WIG	5	0.0005	0.085

C.4 Stable Diffusion class unlearning

In this experiment, we tuned WIG's initialization value p within the range of (0.1, 0.15, 0.2). To ensure a fair comparison with SalUn, we adjusted the number of iterations per epoch to three times the length of the forget loader, as SalUn requires three times more forward passes per iteration. The learning rate and number of epochs were fixed at 1e-5 and 5, respectively.

D Additional result of different experiments

This section presents the results of our unlearning experiments across the models and datasets from sections C.1 and C.2. The results for the experiments in Section C.1 are presented in Tables 1–22, which utilize membership inference attacks, and in Tables 23–28, which provide an inter-class confusion test evaluation. The results for section C.2 are detailed in Tables 29–46.

Table 17: Results of the random data forgetting scenario for the ResNet-18 model on the CIFAR-10 dataset. "()" denotes the performance gap against the Retrain.

Method	*MIA loss*	*MIA entropy*	*MIA confidence*	D^F *Acc*	*Avg Gap*	D^R *Acc*	D^{Te} *Acc*
Retrain	49.28±0.25	61.62±0.68	37.37±0.36	82.33±0.10	0.00	100.00±0.00	81.79±0.20
Finetune	66.00±0.50(16.7)	91.48±0.85(29.9)	2.07±0.53(35.3)	100.00±0.00(17.7)	24.89	**100.00±0.00**	82.44±0.06
GA	65.35±0.47(16.1)	98.05±0.53(36.4)	0.37±0.21(37.0)	99.98±0.02(17.7)	26.79	**100.00±0.00**	82.28±0.15
NegGrad	66.16±0.35(16.9)	91.84±0.48(30.2)	1.52±0.34(35.8)	100.00±0.00(17.7)	25.15	**100.00±0.00**	82.38±0.14
CFK	66.34±0.25(17.1)	96.53±0.25(34.9)	0.35±0.13(37.0)	100.00±0.00(17.7)	26.67	**100.00±0.00**	82.34±0.10
EUK	52.89±0.44(3.6)	50.62±8.20(11.0)	18.16±2.97(19.2)	84.08±1.07(1.8)	8.89	84.92±0.24	73.59±0.18
Bad-T	59.27±0.61(10.0)	64.14±0.88(2.5)	18.07±1.00(19.3)	98.09±0.12(15.8)	11.89	99.78±0.01	75.82±0.14
SCRUB+R	55.64±0.64(6.4)	69.04±3.13(7.4)	19.56±1.53(17.8)	91.87±0.95(9.5)	10.28	98.89±1.24	78.43±0.62
SSD	61.33±3.69(12.0)	86.17±10.45(24.6)	7.71±6.21(29.7)	95.88±4.88(13.5)	19.95	96.51±4.03	79.10±2.84
SalUn	**49.02±0.63**(0.3)	54.18±2.21(7.4)	43.12±2.45(5.8)	**83.20±0.36**(0.9)	3.58	99.91±0.01	81.95±0.31
WIG	52.06±0.37(2.8)	**60.18±0.28**(1.4)	**34.34±0.25**(3.0)	87.88±0.27(5.5)	**3.20**	**100.00±0.00**	**83.64±0.31**

Table 18: Results of the random data forgetting scenario for the ResNet-18 model on the CIFAR-100 dataset. "()" denotes the performance gap against the Retrain.

Method	*MIA loss*	*MIA entropy*	*MIA confidence*	D^F *Acc*	*Avg Gap*	D^R *Acc*	D^{Te} *Acc*
Retrain	50.14±0.49	53.02±1.06	46.16±1.21	71.93±0.55	0.00	99.95±0.01	72.17±0.26
Finetune	69.98±0.45(19.8)	85.85±0.36(32.8)	13.48±0.66(32.7)	99.87±0.03(27.9)	28.32	99.99±0.00	73.57±0.24
GA	65.43±0.49(15.3)	89.22±1.88(36.2)	8.15±0.22(38.0)	97.62±0.25(25.7)	28.80	99.24±0.14	71.12±0.02
NegGrad	69.80±0.40(19.7)	86.15±0.52(33.1)	13.44±0.95(32.7)	99.83±0.07(27.9)	28.35	**99.98±0.01**	**73.62±0.31**
CFK	69.59±0.38(19.5)	88.53±0.22(35.5)	10.78±0.41(35.4)	99.84±0.08(27.9)	29.56	**99.98±0.00**	73.04±0.25
EUK	57.13±0.60(7.0)	58.87±0.86(5.8)	**41.87±1.07**(4.3)	83.88±0.53(11.9)	7.27	99.97±0.00	68.88±0.45
Bad-T	64.77±1.04(14.6)	65.45±1.64(12.4)	16.54±0.66(29.6)	99.45±0.08(27.5)	21.05	99.87±0.01	69.41±0.35
SCRUB+R	57.65±0.28(7.5)	48.60±0.57(4.4)	34.53±0.18(11.6)	84.13±2.23(12.2)	8.94	96.56±1.96	64.04±1.23
SSD	64.41±0.59(14.3)	86.67±4.60(33.6)	8.57±4.09(37.6)	95.46±2.83(23.5)	27.26	95.65±2.90	69.50±2.09
SalUn	56.31±0.52(6.2)	48.36±3.57(4.7)	39.67±3.53(6.5)	89.77±0.85(17.8)	8.79	99.78±0.13	71.35±0.21
WIG	**55.48±0.38**(5.3)	**55.14±1.52**(2.1)	41.60±0.43(4.6)	**75.23±0.14**(3.3)	**3.83**	99.12±0.16	63.43±0.68

Table 19: Results of the random data forgetting scenario for the VGG-16 model on the CIFAR-10 dataset. "()" denotes the performance gap against the Retrain.

Method	*MIA loss*	*MIA entropy*	*MIA confidence*	D^F *Acc*	*Avg Gap*	D^R *Acc*	D^{Te} *Acc*
Retrain	49.53±0.61	68.12±0.45	30.04±0.67	82.36±0.28	0.00	100.00±0.00	82.46±0.15
Finetune	60.10±3.19(10.6)	90.00±6.59(21.9)	5.19±5.55(24.8)	98.41±2.23(16.0)	18.34	**99.92±0.12**	**83.80±0.69**
GA	59.71±3.14(10.2)	94.69±5.43(26.6)	7.46±9.94(22.6)	93.88±8.50(11.5)	17.71	94.28±8.09	78.09±7.03
NegGrad	52.24±2.98(2.7)	83.97±20.22(15.8)	34.25±36.06(4.2)	38.63±38.77(43.7)	16.62	41.01±40.73	34.29±31.41
CFK	62.83±0.75(13.3)	97.98±0.57(29.9)	0.46±0.33(29.6)	100.00±0.00(17.6)	22.59	100.00±0.00	83.15±0.51
EUK	58.85±0.29(9.3)	81.60±1.97(13.5)	5.80±0.61(24.2)	99.64±0.18(17.3)	16.08	99.66±0.22	81.73±0.43
Bad-T	54.73±0.68(5.2)	66.19±4.68(1.9)	**28.03±5.19**(2.0)	99.66±0.19(17.3)	6.61	99.97±0.04	82.54±0.61
SCRUB+R	54.80±1.09(5.3)	67.49±1.98(0.6)	24.89±4.15(5.1)	91.97±1.71(9.6)	5.17	99.20±1.10	81.96±1.08
SSD	62.45±0.81(12.9)	98.38±0.78(30.3)	0.51±0.55(29.5)	99.97±0.05(17.6)	22.58	99.96±0.05	82.98±0.43
SalUn	51.71±0.65(2.2)	**68.20±0.59**(0.1)	27.77±1.23(2.3)	87.38±1.29(5.0)	2.39	99.83±0.23	83.23±1.05
WIG	**50.95±0.42**(1.4)	68.23±0.32(0.1)	26.33±1.98(3.7)	**83.84±2.26**(1.5)	**1.68**	96.76±1.65	81.14±0.91

Table 20: Results of the random data forgetting scenario for the ALL-CNN model on the SVHN dataset. "()" denotes the performance gap against the Retrain.

Method	*MIA loss*	*MIA entropy*	*MIA confidence*	D^F *Acc*	*Avg Gap*	D^R *Acc*	D^{Te} *Acc*
Retrain	50.21±0.33	80.42±0.67	17.55±1.83	94.16±0.59	0.00	99.84±0.23	94.84±0.81
Finetune	52.47±0.68(2.3)	84.65±0.65(4.2)	12.74±1.16(4.8)	97.12±1.31(3.0)	3.57	99.52±0.64	94.14±1.09
GA	54.53±3.04(4.3)	37.81±40.60(42.6)	37.09±42.44(19.5)	41.44±41.29(52.7)	29.80	42.12±40.89	40.52±38.38
NegGrad	52.28±0.27(2.1)	79.61±3.64(0.8)	13.53±3.09(4.0)	96.00±1.98(1.8)	2.19	97.79±2.32	93.06±1.49
CFK	54.44±0.99(4.2)	**81.05±6.49**(0.6)	13.83±4.06(3.7)	97.21±2.25(3.0)	2.91	98.09±1.82	92.09±2.73
EUK	51.48±0.55(1.3)	77.62±3.58(2.8)	20.07±1.87(2.5)	**94.43±0.18**(0.3)	1.72	97.71±0.16	93.32±0.33
Bad-T	53.87±0.36(3.7)	84.47±1.79(4.0)	12.71±0.25(4.8)	100.00±0.00(5.8)	4.60	**100.00±0.00**	**94.91±0.05**
SCRUB+R	53.33±1.89(3.1)	86.21±5.47(5.8)	12.10±4.32(5.5)	97.91±1.70(3.8)	4.53	99.24±0.74	94.46±0.44
SSD	53.02±1.01(2.8)	72.04±8.44(8.4)	23.74±11.26(6.2)	72.15±30.98(22.0)	9.85	72.94±30.36	70.27±26.64
SalUn	**50.19±0.45**(0.0)	71.61±5.41(8.8)	19.16±0.46(1.6)	94.82±0.22(0.7)	2.78	97.81±0.18	94.47±0.09
WIG	50.24±0.17(0.0)	78.19±2.14(2.2)	**16.31±0.45**(1.2)	92.96±0.49(1.2)	**1.18**	98.88±0.34	93.63±0.29

Table 21: Results of the random data forgetting scenario for the ResNet-18 model on the SVHN dataset. "()" denotes the performance gap against the Retrain.

Method	*MIA loss*	*MIA entropy*	*MIA confidence*	D^F *Acc*	*Avg Gap*	D^R *Acc*	D^{Te} *Acc*
Retrain	49.87±0.63	85.58±1.41	12.98±0.40	94.33±0.48	0.00	99.99±0.01	94.67±0.29
Finetune	53.73±2.54(3.9)	89.29±4.96(3.7)	6.83±4.09(6.2)	97.19±1.99(2.9)	4.15	99.26±0.55	93.32±1.42
GA	57.16±0.48(7.3)	98.95±0.24(13.4)	0.20±0.08(12.8)	99.99±0.01(5.7)	9.78	**100.00±0.00**	**95.14±0.03**
NegGrad	54.24±2.29(4.4)	89.01±7.19(3.4)	6.56±4.71(6.4)	97.75±1.80(3.4)	4.41	99.13±0.95	93.74±1.05
CFK	57.82±0.03(8.0)	98.40±0.22(12.8)	0.44±0.12(12.5)	100.00±0.00(5.7)	9.75	**100.00±0.00**	95.10±0.04
EUK	54.34±0.22(4.5)	79.98±3.79(5.6)	10.86±1.32(2.1)	99.16±0.32(4.8)	4.25	99.21±0.18	93.99±0.20
Bad-T	56.93±0.46(7.1)	**85.78±2.89**(0.2)	10.51±1.39(2.5)	99.31±0.11(5.0)	3.68	99.38±0.14	91.80±0.29
SCRUB+R	52.84±2.09(3.0)	85.17±7.79(0.4)	10.98±5.12(2.0)	96.38±2.73(2.0)	1.86	99.24±0.94	93.31±0.83
SSD	53.38±0.90(3.5)	80.69±9.79(4.9)	12.18±6.16(0.8)	**95.22±2.77**(0.9)	2.52	95.77±2.46	90.40±1.91
SalUn	52.95±0.10(3.1)	84.04±1.40(1.5)	14.51±1.04(1.5)	99.74±0.09(5.4)	2.89	99.98±0.00	94.60±0.15
WIG	**50.16±0.16**(0.3)	84.96±1.92(0.6)	**13.20±1.04**(0.2)	93.08±0.62(1.2)	**0.60**	99.57±0.32	92.97±0.70

Table 22: Results of the random data forgetting scenario for the VGG-16 model on the SVHN dataset. "()" denotes the performance gap against the Retrain.

Method	*MIA loss*	*MIA entropy*	*MIA confidence*	D^F *Acc*	*Avg Gap*	D^R *Acc*	D^{Te} *Acc*
Retrain	49.87±0.55	89.08±0.75	10.29±0.32	93.31±0.12	0.00	99.67±0.18	93.50±0.09
Finetune	52.75±0.45(2.9)	92.33±1.61(3.2)	5.42±1.29(4.9)	97.90±1.13(4.6)	3.90	99.39±0.62	93.39±1.02
GA	53.73±0.74(3.9)	68.09±36.07(21.0)	20.10±20.15(9.8)	79.65±23.42(13.7)	12.08	80.11±23.21	73.43±23.65
NegGrad	53.01±0.17(3.1)	92.90±0.68(3.8)	4.71±0.48(5.6)	98.29±0.24(5.0)	4.38	99.44±0.18	93.08±0.42
CFK	54.94±0.18(5.1)	96.77±0.83(7.7)	1.50±0.33(8.8)	99.69±0.14(6.4)	6.98	99.66±0.17	93.15±0.41
EUK	55.30±0.20(5.4)	**89.09±1.15**(0.0)	8.25±0.51(2.0)	99.39±0.18(6.1)	3.39	99.44±0.20	92.82±0.37
Bad-T	54.90±0.47(5.0)	90.08±1.91(1.0)	4.51±1.29(5.8)	99.00±0.53(5.7)	4.38	99.00±0.51	91.66±1.08
SCRUB+R	53.19±1.63(3.3)	66.16±34.50(22.9)	11.16±5.85(0.9)	76.46±31.63(16.9)	10.99	77.08±31.80	73.22±28.24
SSD	54.18±0.15(4.3)	90.10±3.20(1.0)	5.11±1.24(5.2)	98.58±0.15(5.3)	3.94	98.54±0.34	91.72±0.75
SalUn	52.27±0.27(2.4)	87.44±0.94(1.6)	12.63±0.35(2.3)	99.50±0.12(6.2)	3.14	**99.99±0.00**	**94.47±0.12**
WIG	**50.77±0.23**(0.9)	87.98±0.30(1.1)	**10.44±0.53**(0.2)	**93.58±0.41**(0.3)	**0.60**	99.01±0.23	92.84±0.28

Table 23: Results of the class forgetting scenario for the ALL-CNN model on the CIFAR-10 dataset. "()" denotes the performance gap against the Retrain.

Method	*MIA loss*	*MIA entropy*	*MIA confidence*	D^F *Acc*	*Avg Gap*	D^R *Acc*	D^{Te} *Acc*
Retrain	51.70±0.29	46.03±1.07	100.00±0.00	0.00±0.00	0.00	100.00±0.00	86.96±0.01
Finetune	65.57±1.87(13.9)	6.90±1.52(39.1)	97.79±0.32(2.2)	28.23±2.83(28.2)	20.86	**100.00±0.00**	86.66±0.11
GA	60.60±2.09(8.9)	9.52±0.19(36.5)	96.22±0.04(3.8)	9.31±0.20(9.3)	14.63	92.17±0.04	78.34±0.15
NegGrad	62.33±1.84(10.6)	9.28±0.28(36.8)	99.77±0.04(0.2)	2.48±0.50(2.5)	12.52	99.36±0.15	83.46±0.18
CFK	66.13±2.08(14.4)	6.83±1.48(39.2)	97.49±0.39(2.5)	31.06±2.31(31.1)	21.80	**100.00±0.00**	86.68±0.10
EUK	**51.20±1.50**(0.5)	**47.40±2.10**(1.4)	**100.00±0.00**(0.0)	**0.00±0.00**(0.0)	**0.47**	90.38±0.14	82.36±0.06
Bad-T	63.77±0.87(12.1)	6.62±0.84(39.4)	96.25±0.87(3.8)	31.31±3.22(31.3)	21.64	99.94±0.02	85.33±0.05
SCRUB+R	58.23±1.56(6.5)	32.52±0.78(13.5)	99.95±0.04(0.0)	0.23±0.18(0.2)	5.08	99.92±0.01	85.63±0.14
SSD	50.30±0.96(1.4)	63.42±20.38(17.4)	**100.00±0.00**(0.0)	**0.00±0.00**(0.0)	4.70	65.18±9.34	58.24±7.25
SalUn	54.70±0.54(3.0)	30.85±1.90(15.2)	**100.00±0.00**(0.0)	**0.00±0.00**(0.0)	4.54	**100.00±0.00**	**87.20±0.25**
WIG	50.43±1.13(1.3)	43.02±4.32(3.0)	99.99±0.01(0.0)	0.33±0.36(0.3)	1.16	99.32±0.03	85.79±0.24

Table 24: Results of the class forgetting scenario for the ALL-CNN model on the SVHN dataset. "()" denotes the performance gap against the Retrain.

Method	*MIA loss*	*MIA entropy*	*MIA confidence*	D^F *Acc*	*Avg Gap*	D^R *Acc*	D^{Te} *Acc*
Retrain	51.24±0.42	10.30±0.99	100.00±0.00	0.00±0.00	0.00	100.00±0.00	95.72±0.11
Finetune	53.05±3.74(1.8)	11.08±9.09(0.8)	**100.00±0.00**(0.0)	14.25±15.98(14.2)	4.21	97.54±2.44	93.14±1.96
GA	**50.83±2.52**(0.4)	20.91±9.08(10.6)	93.82±3.82(6.2)	10.84±2.53(10.8)	7.01	95.13±1.42	89.50±0.81
NegGrad	52.01±0.67(0.8)	2.20±1.18(8.1)	**100.00±0.00**(0.0)	0.22±0.31(0.2)	2.27	98.23±2.32	93.28±1.52
CFK	57.86±1.04(6.6)	0.04±0.02(10.3)	**100.00±0.00**(0.0)	47.53±10.41(47.5)	16.10	99.98±0.02	95.50±0.42
EUK	54.23±1.42(3.0)	10.89±2.18(0.6)	**100.00±0.00**(0.0)	**0.00±0.00**(0.0)	0.89	97.81±0.04	93.34±0.27
Bad-T	56.23±0.86(5.0)	0.00±0.00(10.3)	99.99±0.01(0.0)	24.03±3.28(24.0)	9.83	**100.00±0.00**	95.50±0.10
SCRUB+R	53.39±2.64(2.1)	**10.12±1.84**(0.2)	**100.00±0.00**(0.0)	**0.00±0.00**(0.0)	0.58	**100.00±0.00**	**95.66±0.14**
SSD	53.03±2.18(1.8)	28.19±25.70(17.9)	**100.00±0.00**(0.0)	**0.00±0.00**(0.0)	4.92	81.84±12.89	76.35±13.12
SalUn	52.16±0.86(0.9)	6.10±0.55(4.2)	**100.00±0.00**(0.0)	1.45±1.19(1.4)	1.64	99.98±0.01	95.30±0.13
WIG	52.05±2.18(0.8)	11.45±1.10(1.1)	**100.00±0.00**(0.0)	**0.00±0.00**(0.0)	**0.49**	98.16±1.07	93.31±0.87

Table 25: Results of the class forgetting scenario for the ResNet-18 model on the CIFAR-10 dataset. "()" denotes the performance gap against the Retrain.

Method	*MIA loss*	*MIA entropy*	*MIA confidence*	D^F *Acc*	*Avg Gap*	D^R *Acc*	D^{Te} *Acc*
Retrain	50.03±1.89	44.62±1.40	100.00±0.00	0.00±0.00	0.00	100.00±0.00	84.89±0.07
Finetune	70.17±1.30(20.1)	8.37±0.74(36.2)	87.82±0.81(12.2)	71.16±1.26(71.2)	34.93	**100.00±0.00**	84.75±0.06
GA	59.67±0.48(9.6)	21.40±0.37(23.2)	96.46±0.21(3.5)	6.08±0.31(6.1)	10.62	86.65±0.04	73.74±0.35
NegGrad	64.50±0.24(14.5)	16.35±0.16(28.3)	98.59±0.14(1.4)	5.21±0.19(5.2)	12.34	99.19±0.20	81.84±0.25
CFK	70.57±1.19(20.5)	9.58±0.66(35.0)	86.81±0.76(13.2)	74.29±0.37(74.3)	35.77	**100.00±0.00**	84.62±0.06
EUK	**50.57±2.84**(0.5)	31.91±12.92(12.7)	**100.00±0.00**(0.0)	**0.00±0.00**(0.0)	3.31	87.90±0.28	77.04±0.16
Bad-T	69.13±1.39(19.1)	3.73±0.44(40.9)	97.70±0.16(2.3)	29.52±0.53(29.5)	22.95	99.97±0.01	83.73±0.01
SCRUB+R	54.80±0.99(4.8)	**46.40±0.95**(1.8)	**100.00±0.00**(0.0)	**0.00±0.00**(0.0)	1.64	99.99±0.00	84.66±0.08
SSD	53.80±1.50(3.8)	36.36±13.97(8.3)	**100.00±0.00**(0.0)	**0.00±0.00**(0.0)	3.01	77.90±6.22	67.39±4.61
SalUn	56.33±1.78(6.3)	35.23±1.75(9.4)	**100.00±0.00**(0.0)	**0.00±0.00**(0.0)	3.92	**100.00±0.00**	**86.20±0.15**
SHs	48.60±0.64(1.4)	50.51±1.72(5.9)	100.00±0.00(0.0)	**0.00±0.00**(0.0)	1.83	**100.00±0.00**	83.54±0.30
WIG	51.83±1.84(1.8)	42.33±0.94(2.3)	**100.00±0.00**(0.0)	**0.00±0.00**(0.0)	**1.02**	**100.00±0.00**	80.83±0.83

Table 26: Results of the class forgetting scenario for the ResNet-18 model on the CIFAR-100 dataset. "()" denotes the performance gap against the Retrain.

Method	*MIA loss*	*MIA entropy*	*MIA confidence*	D^F *Acc*	*Avg Gap*	D^R *Acc*	D^{Te} *Acc*
Retrain	55.00±0.82	21.08±2.18	100.00±0.00	0.00±0.00	0.00	99.95±0.01	73.00±0.42
Finetune	73.00±2.16(18.0)	**21.75±2.65**(0.7)	78.17±1.53(21.8)	93.75±0.74(93.8)	33.56	**99.98±0.00**	**73.62±0.11**
GA	63.33±1.25(8.3)	5.83±0.31(15.2)	98.50±0.41(1.5)	4.00±0.74(4.0)	7.27	94.60±0.27	65.28±0.33
NegGrad	66.33±0.47(11.3)	3.83±1.12(17.2)	99.58±0.24(0.4)	4.33±0.59(4.3)	8.33	99.44±0.08	69.67±0.19
CFK	71.33±2.49(16.3)	19.50±0.61(1.6)	80.83±0.72(19.2)	87.33±2.09(87.3)	31.10	**99.98±0.00**	73.03±0.33
EUK	49.33±6.85(5.7)	27.92±5.10(6.8)	**100.00±0.00**(0.0)	**0.00±0.00**(0.0)	3.13	99.60±0.06	67.68±0.14
Bad-T	67.00±2.83(12.0)	0.33±0.12(20.8)	99.33±0.12(0.7)	39.75±2.48(39.8)	18.29	99.93±0.00	72.85±0.23
SCRUB+R	**52.67±1.70**(2.3)	8.17±3.32(12.9)	**100.00±0.00**(0.0)	**0.00±0.00**(0.0)	3.81	99.80±0.07	71.69±0.24
SSD	59.00±0.82(4.0)	2.33±1.56(18.8)	**100.00±0.00**(0.0)	**0.00±0.00**(0.0)	5.69	94.70±4.22	67.74±4.16
SalUn	65.00±5.10(10.0)	3.50±0.54(17.6)	99.25±0.54(0.8)	30.58±3.97(30.6)	14.73	**99.98±0.00**	73.47±0.27
WIG	60.33±3.86(5.3)	**21.75±2.68**(0.7)	99.67±0.24(0.3)	2.92±0.85(2.9)	**2.31**	98.89±0.03	66.16±0.20

Table 27: Results of the class forgetting scenario for the ResNet-18 model on the SVHN dataset. "()" denotes the performance gap against the Retrain.

Method	*MIA loss*	*MIA entropy*	*MIA confidence*	D^F *Acc*	*Avg Gap*	D^R *Acc*	D^{Te} *Acc*
Retrain	52.05±1.20	28.11±18.29	100.00±0.00	0.00±0.00	0.00	99.87±0.18	94.64±1.18
Finetune	59.79±0.21(7.7)	1.78±2.50(26.3)	94.04±7.62(6.0)	91.71±4.69(91.7)	32.93	**100.00±0.00**	**95.63±0.07**
GA	49.90±0.30(2.1)	**26.10±3.80**(2.0)	99.85±0.07(0.2)	0.65±0.05(0.7)	**1.24**	97.34±0.71	90.46±0.80
NegGrad	50.40±3.29(1.6)	17.06±0.83(11.1)	**100.00±0.00**(0.0)	0.01±0.01(0.0)	3.18	**100.00±0.00**	94.85±0.09
CFK	58.28±0.17(6.2)	1.91±2.66(26.2)	94.35±6.96(5.7)	92.62±4.00(92.6)	32.68	**100.00±0.00**	95.47±0.02
EUK	54.63±2.37(2.6)	5.28±3.87(22.8)	**100.00±0.00**(0.0)	**0.00±0.00**(0.0)	6.35	90.89±2.38	86.07±2.70
Bad-T	56.31±2.13(4.3)	0.00±0.00(28.1)	**100.00±0.00**(0.0)	10.04±1.56(10.0)	10.60	**100.00±0.00**	95.08±0.06
SCRUB+R	50.59±0.71(1.5)	99.58±0.27(71.5)	0.00±0.00(100.0)	**0.00±0.00**(0.0)	43.23	20.89±0.06	21.56±0.13
SSD	55.04±2.04(3.0)	0.85±1.01(27.3)	99.56±0.46(0.4)	46.63±32.97(46.6)	19.33	99.79±0.30	94.46±1.18
SalUn	**52.01±0.79**(0.0)	6.88±1.40(21.2)	**100.00±0.00**(0.0)	**0.00±0.00**(0.0)	5.32	**100.00±0.00**	95.11±0.17
WIG	54.13±2.31(2.1)	17.09±7.71(11.0)	**100.00±0.00**(0.0)	**0.00±0.00**(0.0)	3.28	96.80±0.36	92.56±0.50

Table 28: Results of the class forgetting scenario for the VGG-16 model on the CIFAR-10 dataset. "()" denotes the performance gap against the Retrain.

Method	*MIA loss*	*MIA entropy*	*MIA confidence*	D^F *Acc*	*Avg Gap*	D^R *Acc*	D^{Te} *Acc*
Retrain	50.73±0.78	47.84±1.57	100.00±0.00	0.00±0.00	0.00	100.00±0.00	85.48±0.20
Finetune	67.30±5.83(16.6)	16.07±10.59(31.8)	80.27±18.36(19.7)	60.53±34.88(60.5)	32.15	99.79±0.30	85.83±0.66
GA	58.77±3.44(8.0)	**48.53±1.81**(0.7)	94.50±1.22(5.5)	7.72±1.41(7.7)	5.49	88.41±1.47	76.00±0.98
NegGrad	62.63±4.02(11.9)	27.68±7.29(20.2)	98.53±1.15(1.5)	7.73±4.70(7.7)	10.32	98.79±0.87	83.37±0.75
CFK	70.40±0.73(19.7)	11.16±8.04(36.7)	67.18±13.18(32.8)	93.96±7.58(94.0)	45.78	**100.00±0.00**	85.16±0.27
EUK	52.57±2.27(1.8)	22.44±4.19(25.4)	**100.00±0.00**(0.0)	**0.00±0.00**(0.0)	6.81	99.81±0.15	84.45±0.35
Bad-T	68.43±0.25(17.7)	0.00±0.00(47.8)	**100.00±0.00**(0.0)	36.25±7.35(36.2)	25.45	99.98±0.03	84.94±0.21
SCRUB+R	**52.37±1.22**(1.6)	45.53±3.60(2.3)	**100.00±0.00**(0.0)	**0.00±0.00**(0.0)	**0.99**	99.91±0.04	85.38±0.37
SSD	59.17±6.06(8.4)	54.34±36.37(6.5)	66.77±47.00(33.2)	33.32±47.12(33.3)	20.37	91.95±11.30	79.07±7.39
SalUn	57.30±1.68(6.6)	17.68±3.02(30.2)	**100.00±0.00**(0.0)	**0.00±0.00**(0.0)	9.18	**100.00±0.00**	**86.95±0.36**
WIG	52.70±2.55(2.0)	50.47±6.11(2.6)	**100.00±0.00**(0.0)	0.08±0.12(0.1)	1.17	98.01±0.99	84.11±0.80

Table 29: Results of the class forgetting scenario for the VGG-16 model on the SVHN dataset. "()" denotes the performance gap against the Retrain.

Method	*MIA loss*	*MIA entropy*	*MIA confidence*	D^F *Acc*	*Avg Gap*	D^R *Acc*	D^{Te} *Acc*
Retrain	51.21±2.41	38.46±2.46	100.00±0.00	0.00±0.00	0.00	99.68±0.08	93.73±0.18
Finetune	53.74±1.12(2.5)	12.23±0.33(26.2)	**100.00±0.00**(0.0)	15.92±1.21(15.9)	11.17	99.69±0.19	93.77±0.26
GA	52.97±1.71(1.8)	49.52±5.35(11.1)	99.83±0.08(0.2)	0.38±0.25(0.4)	3.34	90.12±3.49	84.02±2.91
NegGrad	51.87±2.52(0.7)	**34.59±5.18**(3.9)	**100.00±0.00**(0.0)	**0.00±0.00**(0.0)	**1.13**	99.42±0.27	93.15±0.61
CFK	56.65±0.51(5.4)	1.33±0.87(37.1)	**100.00±0.00**(0.0)	57.47±15.00(57.5)	25.01	100.00±0.00	94.17±0.27
EUK	51.87±2.28(0.7)	12.01±4.75(26.5)	**100.00±0.00**(0.0)	**0.00±0.00**(0.0)	6.78	99.92±0.05	93.89±0.27
Bad-T	52.98±1.10(1.8)	0.00±0.00(38.5)	**100.00±0.00**(0.0)	17.64±12.79(17.6)	14.47	99.31±0.45	93.76±0.92
SCRUB+R	52.45±3.12(1.2)	5.10±2.59(33.4)	**100.00±0.00**(0.0)	**0.00±0.00**(0.0)	8.65	100.00±0.00	**94.71±0.15**
SSD	55.47±0.84(4.3)	49.53±38.06(11.1)	38.88±43.67(61.1)	65.72±46.48(65.7)	35.54	99.29±0.45	92.44±1.03
SalUn	**51.40±0.82**(0.2)	17.94±12.76(20.5)	**100.00±0.00**(0.0)	**0.00±0.00**(0.0)	5.18	**99.99±0.00**	**94.71±0.06**
WIG	51.54±0.99(0.3)	24.55±9.74(13.9)	**100.00±0.00**(0.0)	**0.00±0.00**(0.0)	3.56	98.19±0.16	92.67±0.61

Table 30: Results of the selective forgetting scenario for the ALL-CNN model on the CIFAR-10 dataset. "()" denotes the performance gap against the Retrain.

Method	*MIA loss*	*MIA entropy*	*MIA confidence*	D^F *Acc*	*Avg Gap*	D^R *Acc*	D^{Te} *Acc*
Retrain	54.33±2.05	59.00±2.16	47.33±0.94	73.67±1.25	0.00	100.00±0.00	84.34±0.18
Finetune	69.33±2.62(15.0)	88.33±0.47(29.3)	7.67±0.47(39.7)	100.00±0.00(26.3)	27.58	**100.00±0.00**	84.32±0.29
GA	**55.00±2.94**(0.7)	64.00±3.74(5.0)	88.33±1.25(41.0)	14.67±3.09(59.0)	26.42	90.88±0.70	76.44±0.44
NegGrad	57.00±2.83(2.7)	35.33±3.86(23.7)	85.67±1.25(38.3)	26.67±2.62(47.0)	27.92	95.98±0.61	78.99±0.18
CFK	70.33±3.30(16.0)	94.67±0.94(35.7)	3.33±0.47(44.0)	100.00±0.00(26.3)	30.50	**100.00±0.00**	84.34±0.28
EUK	48.33±4.71(6.0)	50.33±9.18(8.7)	26.33±5.91(21.0)	**74.33±5.44**(0.7)	9.08	82.63±0.17	76.20±0.21
Bad-T	75.00±2.45(20.7)	78.33±9.46(19.3)	25.33±7.72(22.0)	100.00±0.00(26.3)	22.08	**100.00±0.00**	84.20±0.03
SCRUB+R	50.33±2.49(4.0)	72.33±2.05(13.3)	**50.33±1.70**(3.0)	64.33±3.09(9.3)	7.42	99.99±0.00	83.19±0.07
SSD	53.00±8.29(1.3)	51.00±7.79(8.0)	100.00±0.00(52.7)	0.00±0.00(73.7)	33.92	58.70±11.99	51.51±8.77
SalUn	66.33±4.19(12.0)	86.67±2.87(27.7)	14.00±2.16(33.3)	98.33±0.47(24.7)	24.41	**100.00±0.00**	84.34±0.09
WIG	52.33±3.77(2.0)	**56.67±6.34**(2.3)	43.67±6.02(3.7)	81.67±4.50(8.0)	**4.00**	**100.00±0.00**	**84.90±0.34**

Table 31: Results of the selective forgetting scenario for the ALL-CNN model on the SVHN dataset. "()" denotes the performance gap against the Retrain.

Method	*MIA loss*	*MIA entropy*	*MIA confidence*	D^F *Acc*	*Avg Gap*	D^R *Acc*	D^{Te} *Acc*
Retrain	52.33±2.36	83.67±4.92	16.00±4.55	93.00±2.45	0.00	99.92±0.09	94.91±0.71
Finetune	51.67±1.25(0.7)	78.33±7.72(5.3)	**15.67±5.56**(0.3)	96.33±1.89(3.3)	2.41	99.79±0.13	94.02±0.49
GA	58.00±4.90(5.7)	49.00±4.97(34.7)	30.00±4.55(14.0)	100.00±0.00(7.0)	15.34	**100.00±0.00**	94.75±0.07
NegGrad	45.67±2.87(6.7)	17.67±5.91(66.0)	65.67±6.18(49.7)	68.33±1.70(24.7)	36.75	94.45±1.30	90.22±0.55
CFK	55.33±2.62(3.0)	85.33±6.60(1.7)	10.33±3.30(5.7)	99.67±0.47(6.7)	4.25	99.94±0.05	94.57±0.43
EUK	50.33±0.47(2.0)	**82.33±6.94**(1.3)	14.00±4.97(2.0)	94.33±2.62(1.3)	**1.67**	95.46±1.02	91.55±1.18
Bad-T	58.67±4.50(6.3)	85.67±4.50(2.0)	10.33±5.31(5.7)	100.00±0.00(7.0)	5.25	**100.00±0.00**	**95.48±0.14**
SCRUB+R	51.33±2.87(1.0)	78.67±1.70(5.0)	28.00±2.16(12.0)	88.00±1.41(5.0)	5.75	**100.00±0.00**	95.43±0.11
SSD	49.00±5.89(3.3)	4.67±4.64(79.0)	100.00±0.00(84.0)	0.00±0.00(93.0)	64.83	81.30±7.96	76.09±8.41
SalUn	**52.67±6.34**(0.3)	91.33±1.89(7.7)	9.00±2.16(7.0)	99.00±0.00(6.0)	5.25	**100.00±0.00**	95.47±0.11
WIG	50.00±2.16(2.3)	73.00±9.42(10.7)	19.67±6.55(3.7)	**93.33±2.05**(0.3)	4.25	97.15±0.31	93.30±0.49

Table 32: Results of the selective forgetting scenario for the ResNet-18 model on the CIFAR-10 dataset. "()" denotes the performance gap against the Retrain.

Method	*MIA loss*	*MIA entropy*	*MIA confidence*	D^F *Acc*	*Avg Gap*	D^R *Acc*	D^{Te} *Acc*
Retrain	54.33±4.50	53.00±2.83	50.33±4.11	71.33±4.78	0.00	100.00±0.00	82.32±0.04
Finetune	68.33±3.09(14.0)	90.67±3.40(37.7)	1.00±0.82(49.3)	100.00±0.00(28.7)	32.42	**100.00±0.00**	82.44±0.09
GA	72.00±0.82(17.7)	31.67±1.25(21.3)	**49.67±3.40**(0.7)	90.33±4.11(19.0)	14.66	99.72±0.15	80.11±0.10
NegGrad	61.00±9.09(6.7)	64.67±6.34(11.7)	86.00±0.82(35.7)	15.33±1.25(56.0)	27.50	96.72±0.49	77.78±0.18
CFK	68.00±2.94(13.7)	96.33±1.25(43.3)	0.33±0.47(50.0)	100.00±0.00(28.7)	33.92	**100.00±0.00**	82.38±0.10
EUK	58.00±2.16(3.7)	69.00±0.00(16.0)	16.67±0.94(33.7)	90.00±2.16(18.7)	18.00	95.56±0.13	78.94±0.15
Bad-T	71.00±2.83(16.7)	50.67±0.47(2.3)	37.33±2.05(13.0)	100.00±0.00(28.7)	15.17	**100.00±0.00**	81.84±0.14
SCRUB+R	73.67±2.05(19.3)	76.67±2.36(23.7)	75.33±0.47(25.0)	28.33±1.70(43.0)	27.75	99.88±0.03	80.85±0.19
SSD	**54.33±2.36**(0.0)	29.67±7.41(23.3)	100.00±0.00(49.7)	0.00±0.00(71.3)	36.08	81.73±5.18	68.38±3.20
SalUn	49.67±5.73(4.7)	66.00±3.56(13.0)	38.67±5.44(11.7)	81.33±2.87(10.0)	9.83	**100.00±0.00**	82.57±0.09
SHs	50.67±5.56(3.6)	58.33±2.05(5.3)	46.00±2.16(4.3)	**73.00±2.16**(1.6)	**3.74**	**100.00±0.00**	81.03±0.04
WIG	52.33±6.85(2.0)	**53.67±4.03**(0.7)	43.00±4.90(7.3)	81.67±3.86(10.3)	5.09	**100.00±0.00**	**83.98±0.17**

Method	*MIA loss*	*MIA entropy*	*MIA confidence*	D^F *Acc*	*Avg Gap*	D^R *Acc*	D^{Te} *Acc*
Retrain	50.00±5.10	49.33±5.91	46.33±4.03	73.00±1.41	0.00	99.95±0.01	72.81±0.12
Finetune	71.67±0.94(21.7)	87.33±2.62(38.0)	12.67±2.62(33.7)	100.00±0.00(27.0)	30.08	**99.98±0.00**	**73.75±0.16**
GA	76.00±0.82(26.0)	34.67±5.44(14.7)	**48.67±5.56**(2.3)	90.67±0.47(17.7)	15.17	99.79±0.03	71.64±0.22
NegGrad	**53.67±5.44**(3.7)	28.33±1.25(21.0)	94.67±0.47(48.3)	16.67±1.70(56.3)	32.34	99.06±0.16	69.24±0.27
CFK	71.00±0.82(21.0)	86.33±0.47(37.0)	13.33±0.94(33.0)	100.00±0.00(27.0)	29.50	**99.98±0.00**	73.18±0.30
EUK	56.33±2.36(6.3)	46.33±14.08(3.0)	38.00±9.42(8.3)	**76.00±10.42**(3.0)	**5.16**	99.52±0.14	68.00±0.57
Bad-T	73.33±0.94(23.3)	29.33±4.11(20.0)	56.67±1.70(10.3)	94.67±2.05(21.7)	18.84	99.95±0.00	72.86±0.13
SCRUB+R	54.33±6.13(4.3)	44.67±7.93(4.7)	60.67±5.91(14.3)	53.00±7.79(20.0)	10.83	98.99±0.14	69.28±0.23
SSD	55.67±2.87(5.7)	5.67±5.25(43.7)	100.00±0.00(53.7)	0.00±0.00(73.0)	44.00	99.03±0.22	71.87±0.05
SalUn	64.33±3.30(14.3)	83.33±4.03(34.0)	17.67±4.78(28.7)	98.00±0.82(25.0)	25.50	99.97±0.00	73.48±0.14
WIG	64.00±2.94(14.0)	**48.00±5.35**(1.3)	51.33±4.99(5.0)	87.33±2.49(14.3)	8.66	**99.98±0.00**	71.00±0.22

Table 33: Results of the selective forgetting scenario for the ResNet-18 model on the CIFAR-100 dataset. "()" denotes the performance gap against the Retrain.

Table 34: Results of the selective forgetting scenario for the ResNet-18 model on the SVHN dataset. "()" denotes the performance gap against the Retrain.

Method	*MIA loss*	*MIA entropy*	*MIA confidence*	D^F *Acc*	*Avg Gap*	D^R *Acc*	D^{Te} *Acc*
Retrain	50.33±2.62	84.00±0.82	14.33±3.40	92.00±2.45	0.00	99.98±0.03	94.75±0.46
Finetune	54.33±2.05(4.0)	93.00±6.48(9.0)	4.33±4.78(10.0)	97.33±3.77(5.3)	7.08	99.30±0.99	94.29±1.39
GA	57.67±4.11(7.3)	76.67±4.11(7.3)	**14.67±2.87**(0.3)	100.00±0.00(8.0)	5.75	**100.00±0.00**	94.42±0.08
NegGrad	56.67±3.09(6.3)	47.00±5.35(37.0)	33.67±4.19(19.3)	**91.67±1.25**(0.3)	15.75	99.19±0.15	93.07±0.16
CFK	54.00±4.32(3.7)	97.67±1.25(13.7)	0.33±0.47(14.0)	100.00±0.00(8.0)	9.84	**100.00±0.00**	**95.15±0.02**
EUK	57.00±6.48(6.7)	81.00±1.41(3.0)	13.00±1.63(1.3)	98.67±0.47(6.7)	4.42	98.22±0.49	93.05±0.48
Bad-T	56.00±3.56(5.7)	93.00±3.74(9.0)	2.00±1.41(12.3)	100.00±0.00(8.0)	8.75	**100.00±0.00**	95.11±0.01
SCRUB+R	51.00±2.16(0.7)	82.33±2.49(1.7)	15.67±1.25(1.3)	95.00±1.63(3.0)	1.67	**100.00±0.00**	94.97±0.06
SSD	56.33±3.68(6.0)	53.00±36.85(31.0)	32.33±26.96(18.0)	94.67±7.54(2.7)	14.42	99.51±0.67	94.14±0.85
SalUn	51.33±0.94(1.0)	90.00±5.72(6.0)	9.33±5.31(5.0)	99.33±0.47(7.3)	4.83	**100.00±0.00**	95.14±0.10
WIG	**50.00±2.45**(0.3)	**84.00±4.55**(0.0)	13.67±5.44(0.7)	93.00±3.27(1.0)	**0.50**	99.27±0.20	92.75±0.19

Table 35: Results of the selective forgetting scenario for the VGG-16 model on the CIFAR-10 dataset. "()" denotes the performance gap against the Retrain.

Method	*MIA loss*	*MIA entropy*	*MIA confidence*	D^F *Acc*	*Avg Gap*	D^R *Acc*	D^{Te} *Acc*
Retrain	52.00±2.94	58.00±6.68	44.00±7.87	72.00±5.10	0.00	100.00±0.00	82.97±0.11
Finetune	65.33±1.25(13.3)	97.00±0.00(39.0)	0.33±0.47(43.7)	100.00±0.00(28.0)	31.00	**100.00±0.00**	83.16±0.38
GA	73.00±4.32(21.0)	**58.33±6.60**(0.3)	29.67±4.19(14.3)	88.00±2.16(16.0)	12.91	98.80±0.49	80.36±0.29
NegGrad	71.00±3.56(19.0)	45.01±5.72(13.0)	44.33±2.87(0.3)	**73.00±1.41**(1.0)	8.33	97.24±0.24	79.17±0.46
CFK	65.33±1.25(13.3)	97.33±0.47(39.3)	0.33±0.47(43.7)	100.00±0.00(28.0)	31.08	99.98±0.03	83.11±0.42
EUK	69.00±4.08(17.0)	77.67±2.05(19.7)	6.33±2.62(37.7)	99.67±0.47(27.7)	25.50	99.70±0.21	81.68±0.43
Bad-T	69.00±2.45(17.0)	53.00±12.68(5.0)	36.00±14.31(8.0)	99.33±0.94(27.3)	14.33	99.97±0.04	82.78±0.25
SCRUB+R	49.67±1.70(2.3)	66.67±0.47(8.7)	37.00±2.16(7.0)	79.67±0.47(7.7)	6.42	**100.00±0.00**	83.00±0.20
SSD	60.00±8.04(8.0)	75.67±16.68(17.7)	4.67±6.60(39.3)	65.33±46.23(6.7)	17.92	70.33±41.68	58.89±33.58
SalUn	53.33±3.40(1.3)	53.67±6.94(4.3)	43.67±9.46(0.3)	74.33±4.92(2.3)	2.08	**100.00±0.00**	**83.73±0.35**
WIG	**51.33±0.94**(0.7)	58.67±10.21(0.7)	**44.00±15.25**(0.0)	70.67±15.20(1.3)	**0.67**	98.69±0.71	82.56±0.62

Table 36: Results of the selective forgetting scenario for the VGG-16 model on the SVHN dataset. "()" denotes the performance gap against the Retrain.

Method	*MIA loss*	*MIA entropy*	*MIA confidence*	D^F *Acc*	*Avg Gap*	D^R *Acc*	D^{Te} *Acc*
Retrain	49.33±1.25	88.33±6.60	11.00±3.56	92.33±1.70	0.00	99.68±0.19	93.43±0.58
Finetune	54.33±0.47(5.0)	93.00±4.55(4.7)	4.33±3.09(6.7)	98.00±1.63(5.7)	5.50	99.46±0.09	93.03±0.29
GA	56.67±1.25(7.3)	**85.33±4.19**(3.0)	**9.33±2.87**(1.7)	95.33±1.89(3.0)	3.75	98.87±0.59	91.68±1.06
NegGrad	55.67±0.94(6.3)	71.00±0.82(17.3)	20.67±2.05(9.7)	87.00±1.63(5.3)	9.67	98.87±0.07	91.93±0.05
CFK	53.33±1.25(4.0)	95.67±1.25(7.3)	2.33±0.47(8.7)	99.67±0.47(7.3)	6.84	99.92±0.06	93.66±0.28
EUK	56.33±2.49(7.0)	91.67±3.09(3.3)	7.67±2.87(3.3)	99.33±0.47(7.0)	5.17	99.47±0.20	92.91±0.35
Bad-T	55.33±1.25(6.0)	79.67±2.49(8.7)	8.00±0.82(3.0)	98.67±1.25(6.3)	6.00	99.32±0.44	92.45±0.96
SCRUB+R	52.33±1.25(3.0)	78.33±1.25(10.0)	15.33±2.05(4.3)	95.00±0.82(2.7)	5.00	**100.00±0.00**	94.49±0.18
SSD	53.33±2.49(4.0)	60.00±32.78(28.3)	33.67±46.91(22.7)	33.33±47.14(59.0)	28.50	57.46±32.55	54.38±29.91
SalUn	**51.33±1.70**(2.0)	84.33±8.01(4.0)	17.00±6.48(6.0)	**91.33±1.70**(1.0)	3.25	**100.00±0.00**	**94.77±0.14**
WIG	51.67±0.94(2.3)	**85.33±4.50**(3.0)	14.00±2.83(3.0)	**91.33±1.70**(1.0)	**2.34**	98.72±0.28	92.93±0.45

Table 37: Performance overview of Inter-class confusion test on CIFAR-10 using ALL-CNN model.

Method	*Targeted Error* (↓) Forget set	Test set	*Test Accuracy* (↑)
Retrain	54.67±9.57	30.00±1.63	83.66±0.02
Finetune	2823.00±131.23	278.67±25.46	80.44±0.41
GA	1153.33±86.97	188.67±17.02	66.34±0.52
NegGrad	745.00±64.50	95.33±14.20	75.32±0.59
CFK	3104.00±142.98	305.33±27.39	80.05±0.53
EUK	133.00±10.23	52.33±6.94	81.71±0.23
Bad-T	661.67±45.03	104.33±10.78	78.69±0.25
SCRUB	**20.67±15.69**	**4.67±3.40**	56.40±29.26
SSD	797.00±921.56	135.67±156.87	67.27±2.18
SalUn	442.33±83.54	80.00±6.16	**83.50±0.15**
WIG	87.67±11.32	43.67±4.99	83.21±0.10

Table 38: Performance overview of Inter-class confusion test on CIFAR-10 using ResNet-18 model.

Method	*Targeted Error* (↓) Forget set	Test set	*Test Accuracy* (↑)
Retrain	51.67±3.30	30.67±1.89	81.63±0.19
Finetune	3470.67±105.99	297.33±22.95	78.61±0.34
GA	836.33±31.48	112.67±7.13	64.35±0.40
NegGrad	675.33±20.82	69.00±5.72	74.09±0.69
CFK	3772.33±44.06	346.33±19.60	77.94±0.28
EUK	276.33±25.09	89.67±3.30	74.79±0.01
Bad-T	382.00±31.84	63.00±3.27	77.81±0.16
SCRUB	139.67±12.23	31.33±1.70	81.19±0.15
SSD	3357.00±736.19	485.67±157.97	73.22±1.57
SalUn	335.33±40.50	45.67±3.40	**83.29±0.13**
WIG	**83.00±2.16**	**28.00±0.82**	83.20±0.16

Table 39: Performance overview of Inter-class confusion test on CIFAR-10 using VGG-16 model.

Method	*Targeted Error* (↓) Forget set	Test set	*Test Accuracy* (↑)
Retrain	42.33±4.50	23.33±1.25	82.38±0.19
Finetune	3179.33±354.99	278.33±51.53	81.40±0.44
GA	1269.00±139.29	180.00±37.69	67.01±0.41
NegGrad	1139.33±276.31	65.00±24.06	78.19±0.47
CFK	3781.00±99.40	525.00±26.87	77.58±0.34
EUK	970.00±248.95	165.33±26.91	79.39±0.45
Bad-T	1323.67±123.45	70.67±7.59	81.54±0.49
SCRUB	493.00±159.10	85.00±35.39	82.30±1.04
SSD	3586.67±559.22	855.33±215.31	71.56±1.39
SalUn	335.33±42.62	45.00±0.82	**84.46±0.14**
WIG	**52.67±15.63**	**20.67±5.25**	78.58±0.98

Table 40: Performance overview of Inter-class confusion test on SVHN using ALL-CNN model.

Method	*Targeted Error* (↓) Forget set	Test set	*Test Accuracy* (↑)
Retrain	81.33±26.96	35.67±4.50	94.45±0.49
Finetune	1575.33±172.91	227.67±38.03	94.25±0.25
GA	1305.00±152.40	242.00±75.67	76.53±0.80
NegGrad	693.00±23.34	93.00±9.90	90.41±0.39
CFK	1636.67±153.74	236.33±47.15	94.05±0.20
EUK	**72.33±14.52**	**46.33±8.26**	91.43±0.80
Bad-T	2737.33±257.66	538.67±161.08	89.68±1.13
SCRUB	317.33±26.61	85.00±10.20	93.02±0.80
SSD	3702.00±2618.28	1842.67±1320.41	70.86±16.14
SalUn	854.33±156.38	155.00±43.18	**94.41±0.19**
WIG	222.00±88.98	68.67±30.27	91.86±1.51

Table 41: Performance overview of Inter-class confusion test on SVHN using ResNet-18 model.

Method	*Targeted Error* (↓) Forget set	Test set	*Test Accuracy* (↑)
Retrain	105.33±31.59	38.00±12.83	93.88±1.09
Finetune	3163.67±640.14	550.33±143.70	92.40±0.41
GA	1522.67±194.81	322.67±86.14	73.88±0.38
NegGrad	332.33±23.80	**44.33±11.56**	89.40±0.12
CFK	3899.33±44.06	710.33±36.16	91.17±0.07
EUK	1368.00±70.35	281.00±25.47	92.30±0.08
Bad-T	1559.67±254.00	211.33±100.61	91.30±0.26
SCRUB	2061.00±1092.78	1196.67±641.22	18.61±1.07
SSD	2773.67±2204.01	1137.33±804.61	59.59±37.63
SalUn	1161.00±75.66	219.33±33.00	**94.05±0.15**
WIG	**138.00±22.45**	54.00±4.90	91.80±1.12

Table 42: Performance overview of Inter-class confusion test on SVHN using VGG-16 model.

Method	*Targeted Error* (↓) Forget set	Test set	*Test Accuracy* (↑)
Retrain	104.33±4.64	43.67±3.68	92.27±1.01
Finetune	1289.67±201.12	238.33±14.43	93.38±0.24
GA	387.33±204.62	66.67±14.01	72.42±3.81
NegGrad	427.33±29.94	92.00±22.55	89.30±0.57
CFK	3439.33±171.97	687.33±38.78	90.88±0.23
EUK	3059.33±84.18	1296.00±7.26	87.09±0.61
Bad-T	1885.33±182.46	628.00±407.53	88.78±1.76
SCRUB	2930.00±0.00	5099.00±0.00	19.59±0.00
SSD	3434.00±1296.62	1598.67±472.36	83.14±2.40
SalUn	586.67±118.39	122.00±30.23	**94.33±0.21**
WIG	**138.33±36.65**	**53.33±5.91**	91.99±0.37

Table 43: Results of sparsity-aware unlearning of the class forgetting scenario for the ALL-CNN model on the CIFAR-10 dataset. "()" denotes the performance gap against the Retrain.

Method	*MIA loss*	*MIA entropy*	*MIA confidence*	D^F *Acc*	*Avg Gap*	D^R *Acc*	D^{Te} *Acc*
Retrain	52.57±1.86	29.67±2.96	100.00±0.00	0.00±0.00	0.00	97.06±0.11	90.37±0.17
Finetune	**52.87±1.08**(0.3)	**23.38±7.34**(6.3)	**99.97±0.04**(0.0)	**0.52±0.26**(0.5)	**1.79**	94.44±0.61	89.57±0.28
GA	51.17±0.12(1.4)	17.73±1.29(11.9)	97.01±0.74(3.0)	6.49±0.45(6.5)	5.70	88.90±0.71	84.35±0.47
FF	54.10±1.02(1.5)	81.99±1.90(52.3)	8.78±1.46(91.2)	97.86±0.29(97.9)	60.73	**97.70±0.32**	**91.23±0.11**
IU	53.00±0.65(0.4)	20.57±7.58(9.1)	96.38±1.64(3.6)	11.56±3.46(11.6)	6.18	95.53±0.61	90.16±0.35
WIG	52.07±0.29(0.5)	21.93±4.11(7.7)	99.89±0.07(0.1)	1.32±0.21(1.3)	2.42	94.86±0.11	90.35±0.17

Table 44: Results of sparsity-aware unlearning of the random forgetting scenario for the ALL-CNN model on the CIFAR-10 dataset. "()" denotes the performance gap against the Retrain.

Method	*MIA loss*	*MIA entropy*	*MIA confidence*	D^F *Acc*	*Avg Gap*	D^R *Acc*	D^{Te} *Acc*
Retrain	49.96±0.25	76.59±0.54	17.65±0.48	89.46±0.63	0.00	96.34±0.12	89.09±0.22
Finetune	**51.13±0.40**(1.2)	**76.21±2.00**(0.4)	16.07±1.22(1.6)	91.33±1.02(1.9)	**1.25**	93.51±1.28	88.07±1.09
GA	53.47±0.33(3.5)	84.66±1.26(8.1)	8.64±0.10(9.0)	97.49±0.31(8.0)	7.15	**97.65±0.28**	**91.20±0.21**
FF	52.76±0.95(2.8)	78.74±0.54(2.1)	**17.28±0.24**(0.4)	**88.06±0.16**(1.4)	1.68	87.97±0.31	82.34±0.07
IU	53.04±0.25(3.1)	80.53±2.67(3.9)	10.81±0.75(6.8)	95.40±1.06(5.9)	4.95	95.96±0.78	90.08±0.44
WIG	51.46±0.17(1.5)	79.91±3.11(3.3)	13.57±0.93(4.1)	93.41±0.15(4.0)	3.21	96.22±0.20	90.49±0.36

Table 45: Results of sparsity-aware unlearning of the selective forgetting scenario for the ALL-CNN model on the CIFAR-10 dataset. "()" denotes the performance gap against the Retrain.

Method	*MIA loss*	*MIA entropy*	*MIA confidence*	D^F *Acc*	*Avg Gap*	D^R *Acc*	D^{Te} *Acc*
Retrain	58.33±2.62	77.33±7.13	16.67±3.68	90.67±4.19	0.00	96.36±0.10	89.57±0.39
Finetune	57.00±2.16(1.3)	74.33±6.94(3.0)	**14.67±3.68**(2.0)	**92.67±4.19**(2.0)	2.08	93.77±0.10	88.47±0.12
GA	53.00±7.07(5.3)	33.67±2.87(43.7)	54.00±3.74(37.3)	65.33±6.02(25.3)	27.91	93.54±0.16	87.70±0.19
FF	55.67±1.89(2.7)	84.00±0.82(6.7)	8.33±2.05(8.3)	98.33±1.25(7.7)	6.33	**97.72±0.29**	**91.26±0.14**
IU	**58.00±0.82**(0.3)	**77.67±3.30**(0.3)	**14.67±1.25**(2.0)	95.00±1.41(4.3)	**1.75**	97.43±0.25	91.12±0.06
WIG	56.00±1.41(2.3)	80.00±3.74(2.7)	12.00±3.27(4.7)	93.33±2.49(2.7)	3.08	96.35±0.24	90.60±0.24

Table 46: Results of sparsity-aware unlearning of the class forgetting scenario for the ResNet-18 model on the CIFAR-10 dataset. "()" denotes the performance gap against the Retrain.

Method	*MIA loss*	*MIA entropy*	*MIA confidence*	D^F *Acc*	*Avg Gap*	D^R *Acc*	D^{Te} *Acc*
Retrain	52.67±1.82	32.54±0.27	100.00±0.00	0.00±0.00	0.00	99.98±0.01	93.77±0.26
Finetune	53.13±0.46(0.5)	50.42±3.62(17.9)	**100.00±0.00**(0.0)	**0.00±0.00**(0.0)	4.59	85.92±2.49	83.11±2.24
GA	52.27±2.19(0.4)	**28.99±0.15**(3.6)	93.84±0.76(6.2)	11.15±1.09(11.2)	5.31	96.43±0.46	90.42±0.30
FF	54.27±0.66(1.6)	97.95±0.37(65.4)	0.93±0.31(99.1)	99.73±0.17(99.7)	66.45	**99.65±0.25**	**94.39±0.26**
IU	54.93±1.27(2.3)	17.28±2.38(15.3)	96.04±2.81(4.0)	10.62±6.15(10.6)	8.02	97.49±0.99	91.61±1.16
WIG	**52.50±1.68**(0.2)	26.48±1.65(6.1)	**100.00±0.00**(0.0)	**0.00±0.00**(0.0)	**1.56**	98.07±0.27	93.41±0.29

Table 47: Results of sparsity-aware unlearning of the random forgetting scenario for the ResNet-18 model on the CIFAR-10 dataset. "()" denotes the performance gap against the Retrain.

Method	*MIA loss*	*MIA entropy*	*MIA confidence*	D^F *Acc*	*Avg Gap*	D^R *Acc*	D^{Te} *Acc*
Retrain	49.82±0.48	84.08±1.05	13.72±0.66	93.77±0.61	0.00	99.98±0.01	93.40±0.14
Finetune	51.16±0.33(1.3)	74.56±0.70(9.5)	18.09±2.41(4.4)	90.43±1.62(3.3)	4.64	93.37±1.86	88.28±1.61
GA	52.70±1.63(2.9)	89.32±4.25(5.2)	**12.68±8.62**(1.0)	90.73±7.48(3.0)	3.05	91.12±7.31	85.79±6.79
FF	52.86±1.25(3.0)	89.95±0.25(5.9)	11.04±0.75(2.7)	90.07±0.71(3.7)	3.82	89.95±0.55	85.22±0.28
IU	54.33±0.09(4.5)	96.12±0.27(12.0)	2.02±0.17(11.7)	99.27±0.11(5.5)	8.44	**99.36±0.05**	**93.90±0.33**
WIG	**51.15±0.46**(1.3)	**82.92±0.38**(1.2)	11.19±0.42(2.5)	**94.89±0.52**(1.1)	**1.54**	98.09±0.24	92.72±0.21

Table 48: Results of sparsity-aware unlearning of the selective forgetting scenario for the ResNet-18 model on the CIFAR-10 dataset. "()" denotes the performance gap against the Retrain.

Method	*MIA loss*	*MIA entropy*	*MIA confidence*	D^F *Acc*	*Avg Gap*	D^R *Acc*	D^{Te} *Acc*
Retrain	51.67±2.62	85.67±3.09	11.67±2.05	97.00±1.63	0.00	99.98±0.00	93.46±0.14
Finetune	54.67±2.62(3.0)	97.00±2.16(11.3)	1.33±1.25(10.3)	100.00±0.00(3.0)	6.92	99.92±0.05	94.48±0.06
GA	55.33±1.89(3.7)	56.33±4.50(29.3)	30.33±3.68(18.7)	87.00±2.16(10.0)	15.41	98.04±0.16	91.81±0.19
FF	55.00±2.16(3.3)	99.00±0.82(13.3)	0.67±0.47(11.0)	100.00±0.00(3.0)	7.66	99.65±0.24	94.48±0.21
IU	55.33±1.25(3.7)	99.00±0.82(13.3)	0.67±0.47(11.0)	100.00±0.00(3.0)	7.75	**99.47±0.01**	**94.52±0.21**
WIG	**53.67±4.78**(2.0)	**83.33±4.78**(2.3)	**12.67±4.03**(1.0)	**95.00±0.82**(2.0)	**1.84**	95.83±0.45	90.17±0.36

Table 49: Results of sparsity-aware unlearning of the class forgetting scenario for the VGG-16 model on the CIFAR-10 dataset. "()" denotes the performance gap against the Retrain.

Method	*MIA loss*	*MIA entropy*	*MIA confidence*	D^F *Acc*	*Avg Gap*	D^R *Acc*	D^{Te} *Acc*
Retrain	50.40±1.68	36.45±2.74	100.00±0.00	0.00±0.00	0.00	99.96±0.01	93.16±0.22
Finetune	**50.53±1.86**(0.1)	**43.43±26.40**(7.0)	**100.00±0.00**(0.0)	**0.00±0.00**(0.0)	**1.78**	81.30±5.26	78.52±4.88
GA	52.00±1.49(1.6)	51.20±2.55(14.8)	97.51±0.80(2.5)	3.99±1.23(4.0)	5.71	97.35±0.71	90.69±0.77
FF	55.30±0.92(4.9)	93.85±1.07(57.4)	2.85±0.62(97.2)	99.04±0.33(99.0)	64.62	98.73±0.55	91.45±0.33
IU	53.03±1.35(2.6)	23.78±2.05(12.7)	90.19±4.73(9.8)	18.52±6.56(18.5)	10.91	97.65±0.63	90.66±0.56
WIG	51.37±1.65(1.0)	28.29±0.65(8.2)	**100.00±0.00**(0.0)	**0.00±0.00**(0.0)	2.28	**97.73±0.28**	**92.49±0.32**

Table 50: Results of sparsity-aware unlearning of the random forgetting scenario for the VGG-16 model on the CIFAR-10 dataset. "()" denotes the performance gap against the Retrain.

Method	*MIA loss*	*MIA entropy*	*MIA confidence*	D^F *Acc*	*Avg Gap*	D^R *Acc*	D^{Te} *Acc*
Retrain	49.82±0.31	85.58±0.90	13.19±0.77	92.78±0.51	0.00	99.94±0.01	92.41±0.10
Finetune	**50.75±0.38**(0.9)	63.52±11.27(22.1)	23.26±3.73(10.1)	83.02±1.87(9.8)	10.71	85.02±1.04	81.71±0.77
GA	52.50±1.83(2.7)	68.62±37.09(17.0)	17.82±19.14(4.6)	82.10±21.35(10.7)	8.74	82.38±21.45	77.24±18.71
FF	53.41±1.03(3.6)	92.50±3.83(6.9)	10.81±0.53(2.4)	91.47±1.55(1.3)	3.55	91.43±1.48	85.98±1.43
IU	54.30±0.14(4.5)	96.05±0.30(10.5)	2.34±0.28(10.8)	98.99±0.10(6.2)	8.00	**99.05±0.12**	**92.17±0.43**
WIG	51.27±0.20(1.5)	**82.06±0.11**(3.5)	**12.30±0.63**(0.9)	**94.01±0.37**(1.2)	**1.77**	97.72±0.26	91.64±0.23

Table 51: Results of sparsity-aware unlearning of the selective forgetting scenario for the VGG-16 model on the CIFAR-10 dataset. "()" denotes the performance gap against the Retrain.

Method	*MIA loss*	*MIA entropy*	*MIA confidence*	D^F *Acc*	*Avg Gap*	D^R *Acc*	D^{Te} *Acc*
Retrain	51.33±4.78	86.00±2.94	11.33±4.19	94.33±3.30	0.00	99.93±0.01	92.78±0.16
Finetune	**50.33±6.60**(1.0)	**87.00±3.74**(1.0)	**9.33±3.30**(2.0)	96.67±2.05(2.3)	**1.59**	98.32±0.03	91.27±0.22
GA	55.33±2.87(4.0)	75.67±7.36(10.3)	17.33±4.71(6.0)	**92.33±0.47**(2.0)	5.58	98.62±0.39	91.63±0.28
FF	53.33±4.50(2.0)	94.33±3.30(8.3)	4.00±1.41(7.3)	99.00±0.82(4.7)	5.58	98.75±0.54	91.56±0.31
IU	56.33±0.94(5.0)	98.00±0.82(12.0)	1.67±0.47(9.7)	99.33±0.47(5.0)	7.92	**99.37±0.02**	**93.34±0.11**
WIG	50.00±7.12(1.3)	84.67±2.87(1.3)	8.67±3.86(2.7)	97.33±1.25(3.0)	2.08	98.98±0.10	92.31±0.17

Table 52: Results of sparsity-aware unlearning of the class forgetting scenario for the ALL-CNN model on the SVHN dataset. "()" denotes the performance gap against the Retrain.

Method	*MIA loss*	*MIA entropy*	*MIA confidence*	D^F *Acc*	*Avg Gap*	D^R *Acc*	D^{Te} *Acc*
Retrain	55.45±3.41	8.68±0.66	100.00±0.00	0.00±0.00	0.00	100.00±0.00	95.57±0.06
Finetune	51.71±3.24(3.7)	3.67±0.67(5.0)	99.92±0.12(0.1)	21.24±12.18(21.2)	7.52	99.77±0.09	95.24±0.28
GA	**54.33±0.50**(1.1)	**8.78±5.03**(0.1)	**99.99±0.01**(0.0)	**9.35±1.11**(9.3)	**2.65**	95.75±0.69	94.13±0.62
FF	51.17±0.85(4.3)	76.49±3.02(67.8)	12.06±2.40(87.9)	97.69±0.50(97.7)	64.43	97.87±0.09	95.51±0.24
IU	49.49±1.48(6.0)	3.15±2.46(5.5)	82.94±7.64(17.1)	65.18±4.94(65.2)	23.43	**97.64±0.21**	**95.55±0.22**
WIG	52.13±1.27(3.3)	4.63±0.63(4.0)	99.29±0.45(0.7)	33.74±6.12(33.7)	10.45	99.65±0.01	95.39±0.12

Table 53: Results of sparsity-aware unlearning of the random forgetting scenario for the ALL-CNN model on the SVHN dataset. "()" denotes the performance gap against the Retrain.

Method	*MIA loss*	*MIA entropy*	*MIA confidence*	D^F *Acc*	*Avg Gap*	D^R *Acc*	D^{Te} *Acc*
Retrain	49.84±0.50	85.12±0.46	16.46±0.76	94.71±0.13	0.00	100.00±0.00	95.21±0.11
Finetune	49.30±0.51(0.5)	65.41±5.57(19.7)	22.81±2.57(6.3)	91.68±0.42(3.0)	7.41	93.05±0.65	91.86±1.12
GA	52.20±0.46(2.4)	**82.26±1.04**(2.9)	8.94±0.21(7.5)	98.18±0.17(3.5)	4.05	**98.22±0.08**	**95.84±0.21**
FF	51.93±0.24(2.1)	79.71±1.66(5.4)	10.47±0.61(6.0)	97.81±0.21(3.1)	4.15	97.87±0.07	95.48±0.27
IU	52.17±0.14(2.3)	81.25±0.36(3.9)	9.09±0.20(7.4)	98.01±0.18(3.3)	4.22	98.10±0.08	95.79±0.19
WIG	**49.95±0.90**(0.1)	80.65±1.54(4.5)	**12.71±0.49**(3.8)	**95.57±0.10**(0.9)	**2.30**	99.01±0.18	95.42±0.14

Table 54: Results of sparsity-aware unlearning of the selective forgetting scenario for the ALL-CNN model on the SVHN dataset. "()" denotes the performance gap against the Retrain.

Method	*MIA loss*	*MIA entropy*	*MIA confidence*	D^F *Acc*	*Avg Gap*	D^R *Acc*	D^{Te} *Acc*
Retrain	49.00±2.16	86.33±1.25	14.00±1.63	96.67±1.25	0.00	100.00±0.00	95.21±0.08
Finetune	54.00±4.08(5.0)	85.33±1.70(1.0)	5.00±1.41(9.0)	100.00±0.00(3.3)	4.58	**98.39±0.08**	**95.88±0.16**
GA	52.00±5.10(3.0)	27.00±4.97(59.3)	50.33±4.78(36.3)	83.00±2.94(13.7)	28.08	96.80±0.28	94.88±0.11
FF	**51.00±4.90**(2.0)	**86.00±2.16**(0.3)	6.00±2.45(8.0)	100.00±0.00(3.3)	3.41	98.16±0.10	95.75±0.23
IU	52.33±3.40(3.3)	86.67±3.09(0.3)	7.67±1.25(6.3)	99.00±0.82(2.3)	3.08	98.09±0.07	95.76±0.21
WIG	51.33±5.44(2.3)	83.33±4.50(3.0)	**8.67±3.86**(5.3)	**98.33±1.70**(1.7)	**3.08**	97.61±0.41	95.40±0.28

Table 55: Results of sparsity-aware unlearning of the class forgetting scenario for the ResNet-18 model on the SVHN dataset. "()" denotes the performance gap against the Retrain.

Method	*MIA loss*	*MIA entropy*	*MIA confidence*	D^F *Acc*	*Avg Gap*	D^R *Acc*	D^{Te} *Acc*
Retrain	52.61±1.85	4.58±0.66	100.00±0.00	0.00±0.00	0.00	100.00±0.00	95.87±0.14
Finetune	52.26±4.03(0.4)	3.49±0.67(1.1)	**100.00±0.00**(0.0)	**0.00±0.00**(0.0)	0.36	96.29±0.49	92.85±0.26
GA	**52.61±1.61**(0.0)	19.78±4.56(15.2)	99.19±0.58(0.8)	13.82±3.84(13.8)	7.46	98.94±0.31	93.95±0.47
FF	56.39±0.81(3.8)	96.21±1.43(91.6)	1.32±0.25(98.7)	99.85±0.11(99.8)	73.48	99.68±0.22	95.50±0.11
IU	58.41±0.39(5.8)	17.12±13.30(12.5)	59.94±29.18(40.1)	91.12±11.43(91.1)	37.38	**99.55±0.02**	**95.73±0.10**
WIG	51.94±0.33(0.7)	**4.70±0.34**(0.1)	**100.00±0.00**(0.0)	**0.00±0.00**(0.0)	**0.20**	99.99±0.01	95.65±0.07

Table 56: Results of sparsity-aware unlearning of the random forgetting scenario for the ResNet-18 model on the SVHN dataset. "()" denotes the performance gap against the Retrain.

Method	*MIA loss*	*MIA entropy*	*MIA confidence*	D^F *Acc*	*Avg Gap*	D^R *Acc*	D^{Te} *Acc*
Retrain	49.99±0.34	82.57±0.08	14.30±0.20	95.18±0.26	0.00	100.00±0.00	95.61±0.07
Finetune	55.72±1.17(5.7)	98.53±0.55(16.0)	0.79±0.51(13.5)	99.71±0.21(4.5)	9.93	**99.87±0.09**	**95.67±0.07**
GA	54.87±1.32(4.9)	96.18±1.42(13.6)	2.09±1.15(12.2)	99.19±0.57(4.0)	8.68	99.59±0.29	95.09±0.37
FF	55.49±0.82(5.5)	97.49±0.37(14.9)	1.37±0.04(12.9)	99.70±0.22(4.5)	9.47	99.70±0.21	95.55±0.09
IU	54.30±0.09(4.3)	98.58±0.13(16.0)	1.19±0.07(13.1)	99.44±0.03(4.3)	9.42	99.57±0.01	95.62±0.03
WIG	**51.21±0.18**(1.2)	**84.84±1.61**(2.3)	**10.71±0.24**(3.6)	**94.27±1.19**(0.9)	**2.00**	98.12±0.64	92.29±0.62

Table 57: Results of sparsity-aware unlearning of the selective forgetting scenario for the ResNet-18 model on the SVHN dataset. "()" denotes the performance gap against the Retrain.

Method	*MIA loss*	*MIA entropy*	*MIA confidence*	D^F *Acc*	*Avg Gap*	D^R *Acc*	D^{Te} *Acc*
Retrain	48.33±1.25	81.33±2.62	13.00±0.82	97.00±1.63	0.00	100.00±0.00	95.69±0.03
Finetune	54.67±3.09(6.3)	96.67±0.94(15.3)	1.33±1.25(11.7)	99.67±0.47(2.7)	9.01	99.99±0.01	95.67±0.03
GA	56.67±3.40(8.3)	26.00±7.48(55.3)	32.33±4.71(19.3)	98.33±1.25(1.3)	21.08	99.58±0.25	94.87±0.05
FF	53.00±0.82(4.7)	96.33±1.70(15.0)	1.33±0.47(11.7)	99.67±0.47(2.7)	8.50	99.70±0.21	95.54±0.09
IU	52.33±0.47(4.0)	97.00±0.82(15.7)	1.00±0.00(12.0)	99.00±0.00(2.0)	8.42	**99.57±0.01**	**95.69±0.07**
WIG	**50.67±0.94**(2.3)	**89.00±1.63**(7.7)	**10.67±1.25**(2.3)	**97.33±0.47**(0.3)	**3.17**	99.97±0.01	94.07±0.02

Table 58: Results of sparsity-aware unlearning of the class forgetting scenario for the VGG-16 model on the SVHN dataset. "()" denotes the performance gap against the Retrain.

Method	*MIA loss*	*MIA entropy*	*MIA confidence*	D^F *Acc*	*Avg Gap*	D^R *Acc*	D^{Te} *Acc*
Retrain	50.30±0.93	12.59±0.58	100.00±0.00	0.00±0.00	0.00	100.00±0.00	95.58±0.07
Finetune	54.25±3.43(4.0)	**6.50±3.64**(6.1)	**100.00±0.00**(0.0)	**0.00±0.00**(0.0)	**2.51**	95.46±1.27	91.86±1.90
GA	**49.55±1.12**(0.8)	24.79±3.88(12.2)	73.43±8.32(26.6)	47.69±7.01(47.7)	21.80	97.42±0.63	93.96±0.45
FF	52.34±0.98(2.0)	88.75±1.72(76.2)	4.21±0.68(95.8)	99.30±0.08(99.3)	68.32	99.01±0.17	95.28±0.15
IU	51.94±0.55(1.6)	75.62±11.49(63.0)	11.02±6.14(89.0)	98.11±1.25(98.1)	62.94	99.06±0.09	95.66±0.17
WIG	54.48±2.26(4.2)	5.64±0.59(7.0)	**100.00±0.00**(0.0)	**0.00±0.00**(0.0)	2.78	**99.82±0.00**	**95.76±0.11**

Table 59: Results of sparsity-aware unlearning of the random forgetting scenario for the VGG-16 model on the SVHN dataset. "()" denotes the performance gap against the Retrain.

Method	*MIA loss*	*MIA entropy*	*MIA confidence*	D^F *Acc*	*Avg Gap*	D^R *Acc*	D^{Te} *Acc*
Retrain	50.10±0.46	87.62±0.36	10.50±0.37	95.03±0.25	0.00	100.00±0.00	95.47±0.18
Finetune	53.16±0.05(3.1)	93.21±1.23(5.6)	2.98±0.56(7.5)	99.19±0.21(4.2)	5.08	**99.28±0.13**	**95.71±0.12**
GA	53.15±0.15(3.0)	92.55±1.84(4.9)	3.48±0.70(7.0)	98.81±0.26(3.8)	4.69	99.18±0.16	95.55±0.19
FF	53.27±0.33(3.2)	**89.12±2.22**(1.5)	4.41±1.14(6.1)	99.02±0.23(4.0)	3.69	99.03±0.16	95.36±0.13
IU	53.16±0.25(3.1)	93.58±1.14(6.0)	2.90±0.50(7.6)	99.02±0.08(4.0)	5.15	99.09±0.09	95.65±0.15
WIG	**51.98±0.13**(1.9)	85.90±1.50(1.7)	**8.07±0.90**(2.4)	**97.18±0.38**(2.2)	**2.04**	97.73±0.17	93.89±0.16

Table 60: Results of sparsity-aware unlearning of the selective forgetting scenario for the VGG-16 model on the SVHN dataset. "()" denotes the performance gap against the Retrain.

Method	*MIA loss*	*MIA entropy*	*MIA confidence*	D^F *Acc*	*Avg Gap*	D^R *Acc*	D^{Te} *Acc*
Retrain	49.00±2.94	90.67±0.94	5.67±1.89	97.67±0.47	0.00	99.99±0.00	95.57±0.13
Finetune	46.00±2.45(3.0)	80.00±1.41(10.7)	12.67±2.49(7.0)	96.33±1.70(1.3)	5.50	96.18±0.58	93.19±0.55
GA	**49.00±3.56**(0.0)	71.67±0.47(19.0)	14.33±2.62(8.7)	95.33±0.47(2.3)	7.50	98.76±0.19	95.28±0.13
FF	51.33±1.25(2.3)	93.33±1.25(2.7)	2.00±0.82(3.7)	100.00±0.00(2.3)	2.75	**99.20±0.15**	**95.67±0.13**
IU	**49.00±1.41**(0.0)	93.67±0.47(3.0)	2.33±0.47(3.3)	99.67±0.47(2.0)	2.08	99.09±0.09	95.66±0.15
WIG	49.33±1.70(0.3)	**90.00±2.16**(0.7)	**4.33±1.89**(1.3)	**98.33±0.47**(0.7)	**0.75**	99.79±0.00	95.62±0.03