ADVANCING MACHINE UNLEARNING EVALUATION REQUIRES RETHINKING RETRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Machine unlearning is emerging as a pivotal paradigm aimed at reducing the impact of unwanted data points previously used in training machine learning models. Despite recent advancements in developing new machine unlearning algorithms, robustly evaluating the effectiveness of these algorithms remains a challenging problem. Retraining a model from scratch, which many algorithms seek to circumvent, continues to be a popular benchmark for comparison due to its inherent nature of excluding undesirable data from the outset. In this study, we critically assess the common practice of using a single retrained checkpoint as a point-estimate reference for evaluating machine unlearning algorithms. Through extensive experiments across various evaluation metrics, datasets, and model architectures, we uncover notable instability in retrained models. Our findings reveal that retrained models can exhibit both the highest and lowest effectiveness in unlearning while maintaining the same utility, and that this phenomenon appears across multiple membership inference attack and distance function metrics. Although our results do not negate the importance of retraining as an unlearning method with guaranteed data removal, they highlight the need for careful, distribution-aware application of retraining in machine unlearning evaluation. Code for reproducing all results will be released upon acceptance.

1 INTRODUCTION

With the advent of deep learning, contemporary deep neural networks are often trained on extensive datasets, which may encompass millions of images (Deng et al., 2009) to billions of tokens (Gao et al., 2020). However, these datasets frequently include data points that are sensitive (Kim et al., 2023b; Carlini et al., 2020; Yao et al., 2023), outdated (Hoelscher-Obermaier et al., 2023), or potentially harmful (Zhou et al., 2023), and directly removing these learned samples from deep neural networks remains a non-trivial task. Machine Unlearning (MU) has emerged as a novel approach which, instead of gathering knowledge from data, focuses on minimizing the impact of certain undesirable data points on a pre-trained model (Nguyen et al., 2022; Shaik et al., 2023; Xu et al., 2023). Despite significant advancements in approximate unlearning methods to enhance efficiency, evaluating the performance of an unlearning algorithm and auditing the final unlearned model remains an exceptionally challenging issue (Thudi et al., 2022b; Goel et al., 2022; Kurmanji et al., 2023a).

While MU primarily aims to eliminate the need for retraining, retraining nonetheless remains the evaluation standard in this field, often serving as a benchmark for comparing proposed approximate MU methods (Jia et al., 2023; Kurmanji et al., 2023b; Fan et al., 2024; Wu & Harandi, 2025). However, the process of training a deep neural network through loss function optimization is not completely deterministic (Summers & Dinneen, 2021; Gilmer et al., 2021; Johnson & Zhang, 2023). Concurrently, previous studies typically assess MU algorithms by comparing them to a singular instance of a retrained model (albeit with multiple seeds), utilizing relevant metrics to measure unlearning performance and model utility. This methodology raises questions about the robustness of such evaluations from two distinct perspectives: *(i)* the inherent variance in retraining itself, and *(ii)* the stability of evaluation metrics when applied to equivalent retrained models.

In this paper, by examining retraining in a wide variety of settings, we identify fundamental limitations of using a single retrained checkpoint as a point-estimate standard for benchmarking approximate MU algorithms. Our extensive experiments reveal a dual instability phenomenon: retrained models

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

with identical utility can obtain drastically different unlearning metric scores depending on training configurations, while simultaneously, certain widely-used metrics exhibit high variance across these equivalent models. We employ various evaluation metrics, datasets and model architectures in both class-wise and random unlearning settings, and verify that these instability issues manifest robustly across almost all considered settings. We list our main contributions as follows:

- Our study reveals significant instability in machine unlearning evaluation, stemming from two interrelated sources: *(i)* retraining variance—retrained models, although achieving the same utility, can be trained to exhibit different unlearning metric values depending on hyperparameter choices; and *(ii)* metric fragility—certain evaluation metrics (e.g., MIA-R) show high variance across equivalent retrained models, while others (e.g., MIA-F) remain stable, indicating metric-specific design issues.
- We present an extensive examination of retrained models in various settings, disentangling retraining variance from metric instability. This includes experiments across multiple datasets, model architectures, evaluation metrics, and unlearning tasks.
- We translate these findings into practical guidance: retraining should be reported as a small distribution rather than a single point, aligned with the original training recipe when possible, and paired with robust metrics such as MIA-F rather than fragile retrain-dependent scores.

2 PRELIMINARY

The primary objective of machine unlearning (MU) is to eliminate the influence of a specific subset of data points from a pre-trained model. Following the literature (Xu et al., 2023; Nguyen et al., 2022; Shaik et al., 2023), we categorize the unlearning into two distinct settings: *exact* unlearning and *approximate* unlearning, for which we provide formal definitions below.

Definition 2.1 (Exact Unlearning)**.** Given a learning algorithm $A(\cdot)$, the example space $\mathcal{D}$ and the hypothesis space of models $\mathcal{H}$, the process $U_e(\cdot)$ is an exact MU process if and only if $\forall \mathcal{T} \subseteq \mathcal{H}, D_{tr} \in \mathcal{D}, D_f \subset D_{tr}$ can satisfy

$$\mathbb{P}[A(D_{tr} \backslash D_f) \in \mathcal{T}] = \mathbb{P}[U_e(A(D_{tr}), D_{tr}, D_f) \in \mathcal{T}],$$

where D_{tr} and D_f are the training set and forgetting set, respectively.

$U_e(\cdot)$ can be seen as a special case the unlearning process $U(\cdot)$. Given 1) $\boldsymbol{\theta}_{tr}$, 2) $\boldsymbol{\theta}_r$, 3) $\boldsymbol{\theta}_u$ are the parameters of the model trained 1) on D_{tr} by $A(\cdot)$, 2) on $D_r = D_{tr} \backslash D_f$ by $A(\cdot)$, and 3) unlearned by $U(\cdot)$, respectively, exact unlearning implies the identical distribution in model weights Θ, i.e. $\Theta_r = \Theta_u$ as well as in model outputs, i.e. $\mathbb{P}(M(X; \boldsymbol{\theta}_r)) = \mathbb{P}(M(X; \boldsymbol{\theta}_u)), \forall X \subseteq \mathcal{D}$. M is the inference step of a model and the data. Definition 2.1 also aligns with that in Bourtoule et al. (2021).

Definition 2.2 ((ϵ, δ)-Approximate Unlearning)**.** For a fixed training dataset D_{tr}, forget set $D_f \subseteq D_{tr}$ and a randomized learning algorithm $A : \mathcal{D} \to \mathcal{R}$ where $R \in \mathcal{R}$ is the output region, an approximate unlearning algorithm U is (ϵ, δ)-unlearning with respect to (D_{tr}, D_f, A) if for all regions $R \subseteq \mathcal{R}$, we have

$$\mathbb{P}[A(D_r) \in R] \leq e^{\epsilon} \mathbb{P}[U(A(D_{tr}), D_f, D_{tr}) \in R] + \delta$$
$$\mathbb{P}[U(A(D_{tr}), D_f, D_{tr}) \in R] \leq e^{\epsilon} \mathbb{P}[U(A(D_r) \in R] + \delta$$

Here, D_f can be a subset of the samples from the pre-training data, or simply a single sample $z \in D_f$, indicating the instance-wise unlearning (Cha et al., 2024).

When ϵ and δ are small, $\mathbb{P}[A(D_r)]$ and $\mathbb{P}[U(A(D_{tr}), D_f, D_{tr})]$ are nearly indistinguishable from one another, meaning that the success of unlearning.

3 EXPERIMENTAL SETUP

We evaluate machine unlearning using accuracy, aggregate scores, membership inference attacks, and distance-based metrics. Detailed metric definitions and formulas are provided in Appendix A.2.

Modality	Image	Text
Dataset	CIFAR-10, CIFAR-100, SVHN, Tiny-ImageNet	20Newsgroup
Model architecture	ALLCNN, VGG-16, ResNet-9, ResNet-18, ViT	BERT-base
Learning rate	1e-2, 3e-2, 5e-2, **1e-1**, 2e-1, 3e-1	1e-5, 5e-5, **1e-4**, 1e-3
Epochs (range)	[30, **200**]	[40, **60**]
LR Scheduler	**cosine**, constant	step, **constant**
Weight Decay	1e-4, **3e-4**, 1e-3	**0.0**, 0.01, 0.03

Table 1: The dataset, models, and hyperparameters used across all experiments. The default hyperparmeters used for pre-training are marked in **bold**. All hyperparameter ranges are selected based on the experimental setting in prior work involving the same dataset or model.

Datasets and models To ensure the representativeness of our experiments, we select the most widely used datasets and models from previous work in machine unlearning (Chen et al., 2023; Jia et al., 2023; Fan et al., 2023). We utilize 4 image and 1 text classification datasets: CIFAR-10/100 (Krizhevsky, 2009), SVHN (Netzer et al., 2011), Tiny-ImageNet (Le & Yang, 2015) and 20 Newsgroup (Caldas et al., 2018), respectively. We mainly focus on ResNet-18 (He et al., 2015), ALLCNN (Springenberg et al., 2014), and VGG-16 (Simonyan & Zisserman, 2014) for vision as well as BERT-base (Devlin et al., 2018) for text. The experiment settings are summarized in Table 1.

Pre-training and unlearning We first pre-train the target model on the entire training set. In classwise forgetting, we always try to make the model "forget" class 0, unless specified otherwise. Samples in the forget class are also removed from D_{te} following the previous work (Jia et al., 2023; Fan et al., 2024; Wu & Harandi, 2025). In random forgetting, unless specified otherwise, we randomly sample 10% of the training set without replacement and use the subset as the forget set D_f, and use the remaining 90% as the remain set D_r. The forget set sampling uses a unique seed across all experiments to account for the randomness in the forget sampling procedure, which could add additional variance and mislead our results. However, we do highlight that our main results hold universally across seeds.

Hyperparameter selection To investigate the variations in the space of retrained models, we vary four hyperparameters commonly tuned in practice: learning rate (LR), learning rate schedule, weight decay (WD), and the number of epochs. We conduct a grid search by training a set of retrained models. For each hyperparameter, we define a range based on the choices used in previous work to ensure the selected hyperparameters are reasonable and have been demonstrated to work during retraining. These choices are either reported in the paper or included in the code, ensuring the validity of our selection. Subsequently, we train three models with different and independent seeds for each hyperparameter combination. All hyperparameters are reported in Table 1. We use a wide range of hyperparameters to inspect if the instability can occur on the entire spectrum of retrained models rather than a small cluster of similar retrained models.

Metric implementation details For MIA-based metrics, we evaluate both MIA-F and MIA-R using common attacker choices, including logistic regression and SVM classifiers, and features derived from confidence, entropy, cross-entropy loss, softmax probabilities, and logits. Each fixed-model MIA ablation uses 10 data-splitting seeds and 10 attacker-initialization seeds, yielding 100 attacker runs per configuration. Combined with three independent retraining seeds for each hyperparameter setting, this design separates configuration variance, seed-only retraining variance, and metric-computation variance.

4 Sources of Evaluation Instablity

In this section, we examine the sources of instability in machine unlearning evaluation. We first introduce our visualization approach, then provide an overview of all 10 metrics in both class-wise and random forgetting settings. Subsequently, we systematically disentangle two sources of instability: variance arising from different retraining configurations (retraining variance) and variance arising from metric design choices (metric fragility).

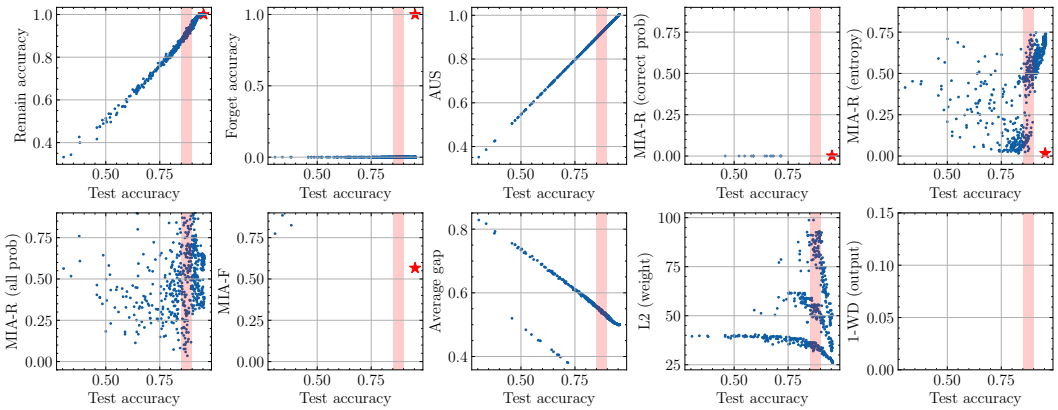


Figure 1: Test accuracy versus ten machine unlearning metrics, including accuracy, MIA, distance, and aggregate metrics, in the **classwise** forgetting using CIFAR-10 with a ResNet-18. Each data point represents a retrained model, generated under a distinct training configuration in Table 1. The accuracy metrics, including remain, forget, and AUS, demonstrate a linear relationship with test accuracy. Notably, significant volatility is discerned in two MIA-R metrics (MIA-R (entropy) and MIA-R (all prob)) and two distance metrics (L2 distance and 1-Wasserstein distance). This volatility is characterized by substantial disparities in the unlearning scores of retrained models, despite these models having comparable test accuracies. The ★ symbol indicates the original pre-trained model (before unlearn). We use the red region to highlight a group of retrained models with similar test accuracy and show per-group standard deviation of metric scores in Figure 4.

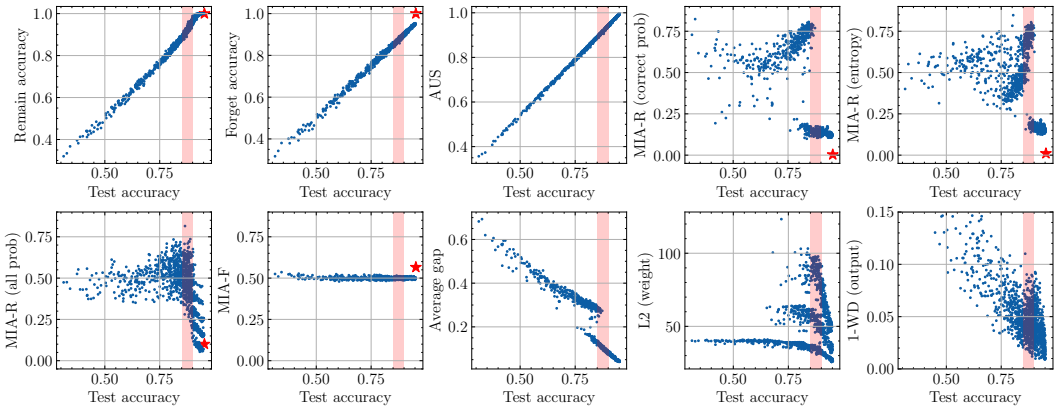


Figure 2: Test accuracy versus ten machine unlearning, including accuracy, MIA, distance, and aggregate metrics, in the context of **random** forgetting using CIFAR-10 with a ResNet-18. Each point corresponds to a retrained model, generated under a distinct training configuration in Table 1. The accuracy metrics, including remain, forget, and AUS, demonstrate a linear relationship with test accuracy. Notably, significant volatility is discerned in all three MIA-R scores and both distance metrics (L2 distance and 1-Wasserstein distance), reflecting the same instability as in the classwise setting. The ★ symbol indicates the original pre-trained model. We use the red region to highlight a group of retrained models with similar test accuracy and show per-group standard deviation of metric scores in Figure 4.

4.1 Visualization Approach

We propose to visualize and quantify the instability of a group of retrained models on the axis of $\mathcal{A}_{te}$, the test accuracy. This is because $\mathcal{A}_{te}$ is an objective metric to evaluate a model's utility. Even after unlearning, we still hope to maintain a high $\mathcal{A}_{te}$ or a score close to the retrained model to avoid the model being useless. Thus, inspecting the target unlearning metric on the axis of $\mathcal{A}_{te}$ allows us to 1)

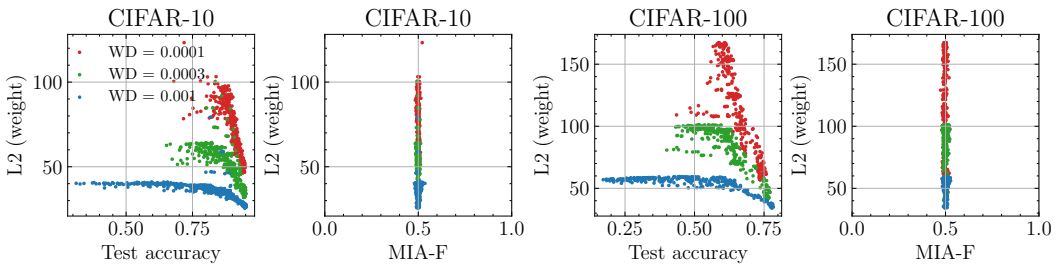


Figure 3: L2 weight distance between retrained and original pre-trained ResNet-18 on CIFAR-10 and CIFAR-100. A strong clustering phenomenon can be observed for all datasets in Row 1, where the L2 distance between two models can be directly controlled by different weight decay values. However, in Row 2, retrained models with the same MIA-R could obta both large and small L2 distance to the original pre-trained model.

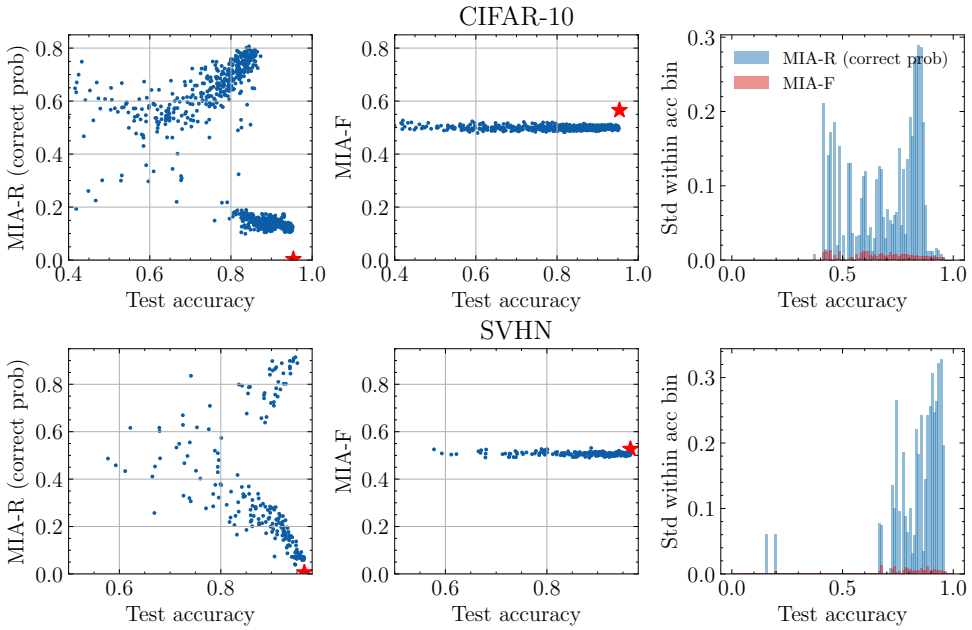
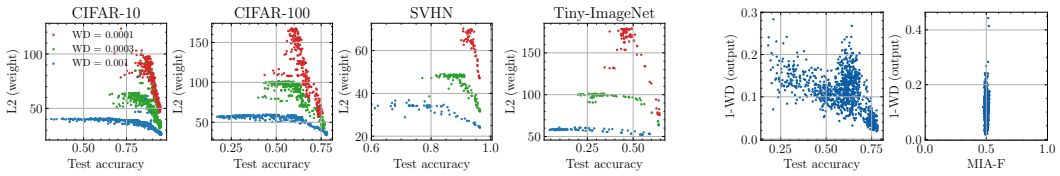


Figure 4: **Row 1 and 2**: MIA-R (correct prob) versus MIA-F of retrained ResNet-18 on CIFAR-10 and SVHN. Retrained models' MIA-R (correct prob) consistently show highly discontinuous MIA scores with phase transitions at multiple stages of the test accuracy. However, retrained models' MIA-F scores stay around 0.5 across all text accuracy. This is the expected behavior as a retrained model trained only on the remain set should not be able to distinguish between the forget set and test set, which are both unseen data. **Row 3**: The per-bin (1%) standard deviation of MIA-R (correct prob) and MIA-F scores calculated from points in Row 1 and Row 2. Each bin represents the standard deviation of MIA scores from a set of retrained models within 1% change in test accuracy.

see how unlearning effectiveness varies against utility and 2) directly observe the variation within a group of retrained models with **similar generalization performance** (i.e., utility).

4.2 Overview of Metric Behaviors

In Figure 1 and Figure 2, we present the results on 10 machine unlearning metrics for a ResNet-18 trained on CIFAR-10. In the first row, we note that both the remaining accuracy, AUS, and average gap exhibit an almost perfect linear relationship with test accuracy. Since the retrained model has not been exposed to samples from the forget class, its forget accuracy remains at zero, consistent with



(a) L2 weight distance. (b) 1-Wasserstein distance.

Figure 5: **Distance-based metrics. (a)** L2 weight distance between retrained and original pre-trained ResNet-18 models. **(b)** 1-Wasserstein distance between the forget set losses and the test set losses from ResNet-18 on CIFAR-100. Both panels show that distance-based metrics can vary substantially among retrained models with similar utility or similar MIA-F scores.

findings in prior studies (Chen et al., 2023; Jia et al., 2023; Fan et al., 2023; 2024; Wu & Harandi, 2025).

MIA-R exhibits substantial instability across retrained models with similar utility. Our results indicate that MIA-R scores from retrained models with the same test accuracy can exhibit considerable instability, in contrast to the consistently high scores of MIA-F. This can be observed via focusing on a single vertical slice of a figure in Figure 1 or Figure 2, which includes MIA scores from a wide range. This instability, as we will demonstrate in Section 5, is the result of both retraining variance and metric design issues.

For MIA-R, we consider training the classifier (attacker) using the correct class probability, the entropy of the predictive distribution, and probabilities from all classes. MIA-R (correct prob) predominantly yields perfect scores. As previously mentioned, since the retrained model is unexposed to the forget class, its output probability for this class will be markedly lower than for other classes. Consequently, it becomes straightforward for the attacker to delineate a boundary between the two distributions. The same justification holds for MIA-F. For MIA-R (entropy) and MIA-R (all prob), the attacker's accuracy varies, ranging between 0 and 0.75, and up to 0.95, respectively. Such variability is undesirable, as the inference of membership should not significantly depend on the retraining method, particularly for retrained models with identical performance. To better see this, we also plot the per-bin standard deviation of the MIA scores in Figure 4. The bin size is 0.01, which means the standard deviation is calculated for every 1% change in test accuracy. We see a high variation in MIA-R scores across the whole range of test accuracy.

Distance metrics exhibit instability primarily driven by retraining configurations. We observe that both the L2 distance in weight and the 1-Wasserstein distance in output space can span a wide range of values. In the case of the L2 distance in weight, we identify three distinct clusters in Figure 1, with each cluster decreasing as test accuracy increases. This phenomenon is later elucidated as being directly related to the weight decay employed during retraining in Figure 3, demonstrating that this variance primarily stems from *retraining variance* rather than *metric fragility*. Similarly, in Figure 5, while no apparent pattern emerges for the 1-Wasserstein distance, the values for models with equivalent performance are dispersed, again reflecting retraining variance.

This conclusion is paralleled in the random forgetting scenario presented in Figure 2. However, in the random setting, all MIA-R variants exhibit instability. Given that the original average gap implementation incorporates MIA-R (correct prob) as one of its components, this instability also impacts the average gap metric, resulting in discontinuities.

5 Disentangling Variance Sources

The instability observed in Figure 1 and Figure 2 could arise from two distinct sources: (i) different retraining configurations producing models with varying unlearning characteristics (retraining variance), or (ii) the evaluation metrics themselves being sensitive to minor variations in model outputs or classifier training randomness (metric fragility). To disentangle these two sources, we conduct a controlled experiment where we condition on a fixed retrained model and examine metric variation; this fixed-model conditioning is repeated across retrained checkpoints when aggregating the ablation results.

5.1 Isolating Metric Fragility

In Figure 6, we study the unlearning performance based on MIA-R and MIA-F using a **fixed retrained model at a time**, thereby eliminating retraining variance within each conditioned analysis. We evaluate ten variants in the type of classifier and the type of features used to train the MIA classifier. Specifically, we employ the two most common linear classifiers used in MIA, logistic regression and SVM. Feature-wise, we take the retrained model's (correct class) confidence, entropy of the softmax output, cross-entropy loss, softmax output, and the logit vector. For each MIA setting, we use 10 data splitting and shuffling seeds and 10 parameter initialization seeds, which results in 100 classifiers per configuration. We report the key statistics of these 100 scores in Figure 6.

MIA-F is robust to both retraining and metric computation randomness. In the lower plot (MIA-F) in Figure 6, MIA-F exhibits consistent test accuracy around 0.5 across all configurations. This demonstrates that MIA-F is a metric robust against both randomness in retraining (as shown in Figures 1 and 2 where MIA-F remains stable across different retrained models) and randomness in MIA classifier training (as shown here with a fixed retrained model).

MIA-R exhibits fragility inherent to its design. In the upper plot of MIA-R in Figure 6, we see drastic fluctuation in MIA test accuracy for softmax and logit features on both logistic regression and SVM classifiers, even when using the same retrained model. This variance cannot be attributed to retraining since the model is fixed—it purely reflects metric design fragility. It is also interesting to see that the SVM classifier with softmax features achieves a maximum of 1.0 and a minimum of 0.0 accuracy. After manually investigating the predictions made by this particular MIA classifier, we find that, while some MIA classifiers have around 0.5 training accuracy (i.e., cannot linearly separate the distributions), the classifier predictions consist of either all 0's or 1's. Given that MIA-R only uses the forget set for evaluation (which consists of all 0's labels, indicating non-membership), predicting all 0's during testing yields an accuracy of 1.0 and vice versa. This binary behavior reflects a fundamental design issue in MIA-R when applied to certain feature types.

5.2 Identifying Retraining Variance

Comparing the variance observed within a single model (Figure 6) to the variance across different retrained models (Figures 1 and 2) allows us to isolate retraining variance. Notably, while MIA-R with confidence, entropy, and loss features exhibit almost no score variation given a fixed retrained model (see the first three pairs in Figure 6), we observe substantial score fluctuation for these same feature types across different retrained models in Figure 1 and Figure 2. This cross-model variance, in the absence of within-model variance, serves as clear evidence that the variation is derived from the optimization (retraining) process rather than from randomness in metric computation.

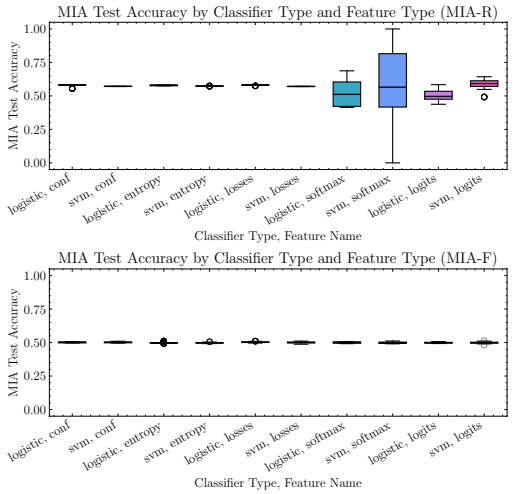


Figure 6: Test accuracy ranges of different MIA-R and MIA-F variants under 100 random seeds.

Similarly, the L2 weight distance clustering pattern observed in Figure 3 directly correlates with the weight decay hyperparameter used during retraining, providing another example of retraining variance. Models trained with different weight decay values occupy distinct clusters in the L2 distance space, despite achieving similar test accuracy. This demonstrates that hyperparameter choices during retraining lead to models at different positions in weight space, which in turn affects distance-based unlearning metrics.

6 DISCUSSION

Our results should not be interpreted as rejecting retraining as an exact unlearning procedure. A model trained directly on $D \backslash D_f$ remains the cleanest reference for data removal. The failure mode we identify is narrower but practically important: evaluation protocols often instantiate this reference with one retrained checkpoint and then treat deviations from that point as evidence for or against an approximate unlearning method. When retrained models with nearly identical utility yield substantially different MIA-R, average-gap, or distance scores, the resulting comparison can depend on which retrained checkpoint was chosen.

This observation is also distinct from the general fact that neural network training is stochastic. Our experiments condition on model utility through test accuracy and show that unlearning-specific metrics can still vary sharply within narrow utility bands. Thus, the instability is not merely that different runs produce different classifiers; rather, commonly used unlearning metrics can assign different privacy or forgetting conclusions to retrained references that are effectively equivalent in utility.

Concrete point-estimate failures The instability remains even when retraining is aligned with the original recipe; full values are reported in Appendix A.5. First, on ResNet-18/CIFAR-10 random forgetting, fixing all hyperparameters to the pre-training recipe and changing only the random seed yields test accuracies between 0.9510 and 0.9519, while MIA-R ranges from 0.321 to 0.449. Thus, the issue is not only an artifact of a broad hyperparameter grid. Second, among utility-matched retrained models whose test accuracy lies in $[0.940, 0.950]$, MIA-R still changes by 0.149. Third, the metric contrast is sharp: under the same original recipe, MIA-F stays in the narrow range 0.5028–0.5069, and under a nearby weight-decay drift it remains 0.5012–0.5062. In contrast, MIA-R is unstable under the original recipe but much tighter under that neighboring drift, indicating that MIA-R depends strongly on the chosen retrained reference while MIA-F remains stable. Finally, the same point-estimate concern appears beyond small image classifiers: on MUSE, a Llama-2-7B text-generation unlearning benchmark (Shi et al., 2024), the retrain-dependent Privacy Leakage score ranges from -0.2671 to 0.3292 across three retrained references, changing even the sign of the metric.

Why this affects current practice The affected metrics are not fringe cases. As summarized in Appendix A.3, MIA-R appears in substantially more recent MU papers than MIA-F, and average gap and distance-based metrics are also retrained-model-dependent. This matters because the accuracy components of average gap are comparatively stable in our experiments, while its MIA component can change sharply with the retrained reference. For example, the 0.128 MIA-R range observed under the original recipe alone can move the MIA term of average gap by about 0.032 before any accuracy component changes. Therefore, the instability of a single retrained checkpoint can propagate into aggregate scores and can potentially alter the ranking of approximate unlearning methods, even when those methods are compared at the same utility level.

7 TOWARDS STABLE UNLEARNING EVALUATION

Our results demonstrate that retraining, when used as a single reference model, is an unstable benchmark for machine unlearning evaluation. Retrained models with indistinguishable utility can exhibit drastically different unlearning metric values due to both optimization randomness and hyperparameter choices, while certain metrics further amplify this variance due to inherent design fragility.

Retraining as a Distribution Rather than treating retraining as a point estimate, we argue that retraining should be viewed as a distribution induced by reasonable training configurations and

random seeds. Similar observations regarding optimization-induced variability have been reported in deep learning more broadly (Summers & Dinneen, 2021; Gilmer et al., 2021). Under this perspective, the effectiveness of an approximate unlearning algorithm should be assessed relative to the variability of retraining itself, for example by comparing the unlearned model's metrics to the confidence interval or interquartile range of retrained models trained on $D \backslash D_f$.

A Tractable Protocol The distributional view need not require hundreds of retrains. A practical protocol is to first follow the original pre-training recipe for retraining on $D \backslash D_f$–using the same optimizer, learning-rate schedule, regularization, and training budget–and then train a small number of reference retrains, e.g., $K = 3$–5, with different seeds. If compute permits, one can additionally probe a small one-factor neighborhood, such as a nearby weight decay or schedule, while retaining only retrained models whose test accuracy lies within a narrow utility window. The reference can then be reported as a median, interquartile range, or confidence interval rather than as one arbitrarily selected checkpoint. This converts retraining from a brittle point estimate into an interval-valued reference at the cost of a few standard baselines.

Amortizing Reference Cost For shared benchmarks, the cost of this protocol can be amortized. Benchmark maintainers can release the retrained checkpoints, raw metric outputs, and reference intervals for common dataset/model/forget-set combinations. A new unlearning method can then be compared against the released reference band without repeating all retraining runs. This is especially useful for larger models, where exhaustive retraining is impractical but a small, aligned set of reference retrains can still expose whether a retrain-dependent metric is stable enough to be used.

Metric Robustness as a Prerequisite Our experiments further show that not all commonly used unlearning metrics are equally reliable. Several MIA-R variants exhibit substantial variance even when evaluated on the same retrained model, indicating fragility intrinsic to metric design rather than retraining randomness. In contrast, MIA-F remains stable across retraining configurations and metric computation randomness, consistent with prior findings on membership inference robustness (Carlini et al., 2022). We therefore recommend treating MIA-F, remain/test/forget accuracy, and AUS as default signals, while using MIA-R and distance-based metrics with caution or as diagnostic evidence rather than primary success criteria.

Scope and Limitations Our main evidence concerns full retraining in discriminative classification, including image classifiers and BERT-based text classification. We analyze evaluation metrics and reference protocols rather than proposing a new unlearning algorithm or a new metric. The same point-estimate concern may arise in larger generative settings whenever a retrain-dependent metric is used, but exhaustive retraining is often infeasible there; in such cases, aligned few-seed reference bands or retrain-free metrics are more practical starting points.

8 Conclusion

Our study reveals fundamental limitations of using a single retrained checkpoint as a benchmark for machine unlearning evaluation. We show that retrained models with equivalent utility can exhibit widely varying unlearning scores across several commonly used metrics, highlighting the need for aligned retraining protocols, distribution-aware references, and robust evaluation metrics.

Impact Statement Machine unlearning supports privacy, compliance, and responsible data governance by enabling models to remove the influence of selected training data. Our work does not introduce new deployment risks; instead, it highlights that unreliable evaluation can misstate whether an unlearning method is effective. More robust, distribution-aware benchmarks can improve trust in unlearning systems by making reported privacy and utility claims less dependent on arbitrary retraining choices.

AI use statement

We used generative AI tools to write the experimental code used in this work. The authors take full responsibility for the code and for the results and conclusions reported in this paper.

REFERENCES

Ali Abbasi, Chayne Thrash, Elaheh Akbari, Daniel Zhang, and Soheil Kolouri. Covarnav: Machine unlearning via model inversion and covariance navigation. *arXiv preprint arXiv:2311.12999*, 2023.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pp. 214–223. PMLR, 2017.

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

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

Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečnỳ, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018.

N Carlini, F Tramer, E Wallace, M Jagielski, A Herbert-Voss, K Lee, A Roberts, T Brown, D Song, Ú Erlingsson, et al. Extracting training data from large language models. arxiv. *Preprint posted online December*, 14:4, 2020.

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

Sungmin Cha, Sungjun Cho, Dasol Hwang, Honglak Lee, Taesup Moon, and Moontae Lee. Learning to unlearn: Instance-wise unlearning for pre-trained classifiers. *arXiv preprint arXiv:2301.11578*, 2023.

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

Jiaao Chen and Diyi Yang. Unlearn what you want to forget: Efficient unlearning for llms. *arXiv preprint arXiv:2310.20150*, 2023.

Kongyang Chen, Yao Huang, and Yiwen Wang. Machine unlearning via gan. *arXiv preprint arXiv:2111.11869*, 2021a.

Kongyang Chen, Yiwen Wang, and Yao Huang. Lightweight machine unlearning in neural network. *arXiv preprint arXiv:2111.05528*, 2021b.

Min Chen, Weizhuo Gao, Gaoyang Liu, Kai Peng, and Chen Wang. Boundary unlearning. *arXiv preprint arXiv:2303.11570*, 2023.

Dasol Choi and Dongbin Na. Towards machine unlearning benchmarks: Forgetting the personal identities in facial recognition systems. *arXiv preprint arXiv:2311.02240*, 2023.

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

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

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

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Zonglin Di, Sixie Yu, Yevgeniy Vorobeychik, and Yang Liu. Adversarial machine unlearning. *arXiv preprint arXiv:2406.07687*, 2024.

Ali Ebrahimpour-Boroojeny, Hari Sundaram, and Varun Chandrasekaran. Not all wrong is bad: Using adversarial examples for unlearning. In *Forty-second International Conference on Machine Learning*.

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

Chongyu Fan, Jiancheng Liu, Alfred Hero, and Sijia Liu. Challenging forgets: Unveiling the worst-case forget sets in machine unlearning. In *European Conference on Computer Vision*, pp. 278–297. Springer, 2024.

Jack Foster, Stefan Schoepf, and Alexandra Brintrup. Fast machine unlearning without retraining through selective synaptic dampening. *arXiv preprint arXiv:2308.07707*, 2023.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

Justin Gilmer, Behrooz Ghorbani, Ankush Garg, Sneha Kudugunta, Behnam Neyshabur, David Cardoze, George Dahl, Zachary Nado, and Orhan Firat. A loss curvature perspective on training instability in deep learning. *arXiv preprint arXiv:2110.04369*, 2021.

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

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

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. 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*, pp. 383–398. Springer, 2020b.

Aditya Golatkar, Alessandro Achille, Avinash Ravichandran, Marzia Polito, and Stefano Soatto. Mixed-privacy forgetting in deep networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 792–801, 2021.

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

Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens Van Der Maaten. Certified data removal from machine learning models. *arXiv preprint arXiv:1911.03030*, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. corr abs/1512.03385 (2015), 2015.

Yingzhe He, Guozhu Meng, Kai Chen, Jinwen He, and Xingbo Hu. Deepobliviate: a powerful charm for erasing data residual memory in deep neural networks. *arXiv preprint arXiv:2105.06209*, 2021.

Jason Hoelscher-Obermaier, Julia Persson, Esben Kran, Ioannis Konstas, and Fazl Barez. Detecting edit failures in large language models: An improved specificity benchmark. *arXiv preprint arXiv:2305.17553*, 2023.

Mark He Huang, Lin Geng Foo, and Jun Liu. Learning to unlearn for robust machine unlearning. In *European Conference on Computer Vision*, pp. 202–219. Springer, 2024a.

Zhehao Huang, Xinwen Cheng, JingHao Zheng, Haoran Wang, Zhengbao He, Tao Li, and Xiaolin Huang. Unified gradient-based machine unlearning with remain geometry enhancement. *Advances in Neural Information Processing Systems*, 37:26377–26414, 2024b.

Zachary Izzo, Mary Anne Smart, Kamalika Chaudhuri, and James Zou. Approximate data deletion from machine learning models. In *International Conference on Artificial Intelligence and Statistics*, pp. 2008–2016. PMLR, 2021.

Jinghan Jia, Jiancheng Liu, Parikshit Ram, Yuguang Yao, Gaowen Liu, Yang Liu, Pranay Sharma, and Sijia Liu. Model sparsification can simplify machine unlearning. *arXiv preprint arXiv:2304.04934*, 2023.

Rie Johnson and Tong Zhang. Inconsistency, instability, and generalization gap of deep neural network training. *arXiv preprint arXiv:2306.00169*, 2023.

Yoonhwa Jung, Ikhyun Cho, Shun-Hsiang Hsu, and Julia Hockenmaier. Attack and reset for unlearning: Exploiting adversarial noise toward machine unlearning through parameter re-initialization. *arXiv preprint arXiv:2401.08998*, 2024.

Hyunjune Kim, Sangyong Lee, and Simon S Woo. Layer attack unlearning: Fast and accurate machine unlearning via layer level attack and knowledge distillation. *arXiv preprint arXiv:2312.16823*, 2023a.

Siwon Kim, Sangdoo Yun, Hwaran Lee, Martin Gubri, Sungroh Yoon, and Seong Joon Oh. Propile: Probing privacy leakage in large language models. *arXiv preprint arXiv:2307.01881*, 2023b.

Sangamesh Kodge, Gobinda Saha, and Kaushik Roy. Deep unlearning: Fast and efficient training-free approach to controlled forgetting. *arXiv preprint arXiv:2312.00761*, 2023.

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

Meghdad Kurmanji, Peter Triantafillou, and Eleni Triantafillou. The brainy student: Scalable unlearning by selectively disobeying the teacher, 2023a. URL https://openreview.net/forum?id=f9eHl5mKx5i.

Meghdad Kurmanji, Peter Triantafillou, and Eleni Triantafillou. Towards unbounded machine unlearning. *arXiv preprint arXiv:2302.09880*, 2023b.

Ya Le and Xuan S. Yang. Tiny imagenet visual recognition challenge. 2015. URL https://api.semanticscholar.org/CorpusID:16664790.

Guihong Li, Hsiang Hsu, Chun-Fu Chen, and Radu Marculescu. Fast-ntk: Parameter-efficient unlearning for large-scale models. *arXiv preprint arXiv:2312.14923*, 2023.

Junde Li and Swaroop Ghosh. Random relabeling for efficient machine unlearning. *arXiv preprint arXiv:2305.12320*, 2023.

Shen Lin, Xiaoyu Zhang, Willy Susilo, Xiaofeng Chen, and Jun Liu. Gdr-gma: Machine unlearning via direction-rectified and magnitude-adjusted gradients. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 9087–9095, 2024.

Yang Liu, Zhuo Ma, Ximeng Liu, Jian Liu, Zhongyuan Jiang, Jianfeng Ma, Philip Yu, and Kui Ren. Learn to forget: Machine unlearning via neuron masking. *arXiv preprint arXiv:2003.10933*, 2020.

Yufang Liu, Yuanbin Wu, and Aimin Zhou. Unlearning with fisher masking. 2022.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

Thanh Tam Nguyen, Thanh Trung Huynh, Phi Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen. A survey of machine unlearning. *arXiv preprint arXiv:2209.02299*, 2022.

Gaurav Patel and Qiang Qiu. Learning to unlearn while retaining: Combating gradient conflicts in machine unlearning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4211–4221, 2025.

Samuele Poppi, Sara Sarto, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Multi-class explainable unlearning for image classification via weight filtering. *arXiv preprint arXiv:2304.02049*, 2023.

Vedant Shah, Frederik Träuble, Ashish Malik, Hugo Larochelle, Michael Mozer, Sanjeev Arora, Yoshua Bengio, and Anirudh Goyal. Unlearning via sparse representations. *arXiv preprint arXiv:2311.15268*, 2023.

Thanveer Shaik, Xiaohui Tao, Haoran Xie, Lin Li, Xiaofeng Zhu, and Qing Li. Exploring the landscape of machine unlearning: A survey and taxonomy. *arXiv preprint arXiv:2305.06360*, 2023.

Jiaeli Shi, Najah Ghalyan, Kostis Gourgoulias, John Buford, and Sean Moran. Deepclean: Machine unlearning on the cheap by resetting privacy sensitive weights using the fisher diagonal. *arXiv preprint arXiv:2311.10448*, 2023.

Weijia Shi, Hsuan-Cheng Lee, Yangsibo Huang, Weijia Kong, Yejin Wang, Huaxiu Wang, Siliang Huang, Tengyu Ma, and Chiyuan Zhang. Muse: Machine unlearning six-way evaluation for language models. *arXiv preprint arXiv:2407.06460*, 2024.

Takashi Shibata, Go Irie, Daiki Ikami, and Yu Mitsuzumi. Learning with selective forgetting. In *IJCAI*, volume 3, pp. 4, 2021.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models (s&p'17). 2016.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

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

Cecilia Summers and Michael J Dinneen. Nondeterminism and instability in neural network optimization. In *International Conference on Machine Learning*, pp. 9913–9922. PMLR, 2021.

Ayush K Tarun, Vikram S Chundawat, Murari Mandal, and Mohan Kankanhalli. Fast yet effective machine unlearning. *IEEE Transactions on Neural Networks and Learning Systems*, 2023a.

Ayush Kumar Tarun, Vikram Singh Chundawat, Murari Mandal, and Mohan Kankanhalli. Deep regression unlearning. In *International Conference on Machine Learning*, pp. 33921–33939. PMLR, 2023b.

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

Anvith Thudi, Hengrui Jia, Ilia Shumailov, and Nicolas Papernot. On the necessity of auditable algorithmic definitions for machine unlearning. In *31st USENIX Security Symposium (USENIX Security 22)*, pp. 4007–4022, 2022b.

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

Jing Wu and Mehrtash Harandi. Scissorhands: Scrub data influence via connection sensitivity in networks. *arXiv preprint arXiv:2401.06187*, 2024.

Jing Wu and Mehrtash Harandi. Munba: Machine unlearning via nash bargaining. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4754–4765, 2025.

Jie Xu, Zihan Wu, Cong Wang, and Xiaohua Jia. Machine unlearning: Solutions and challenges. *arXiv preprint arXiv:2308.07061*, 2023.

Ruikai Yang, Mingzhen He, Zhenghao He, Youmei Qiu, and Xiaolin Huang. Muso: achieving exact machine unlearning in over-parameterized regimes. *Machine Learning*, 114(8):176, 2025.

Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Eric Sun, and Yue Zhang. A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. *arXiv preprint arXiv:2312.02003*, 2023.

Youngsik Yoon, Jinhwan Nam, Hyojeong Yun, Dongwoo Kim, and Jungseul Ok. Few-shot unlearning by model inversion. *arXiv preprint arXiv:2205.15567*, 2022.

Xulong Zhang, Jianzong Wang, Ning Cheng, Yifu Sun, Chuanyao Zhang, and Jing Xiao. Machine unlearning methodology based on stochastic teacher network. In *International Conference on Advanced Data Mining and Applications*, pp. 250–261. Springer, 2023.

Xin Zhou, Yi Lu, Ruotian Ma, Tao Gui, Qi Zhang, and Xuanjing Huang. Making harmful behaviors unlearnable for large language models. *arXiv preprint arXiv:2311.02105*, 2023.

Jianing Zhu, Bo Han, Jiangchao Yao, Jianliang Xu, Gang Niu, and Masashi Sugiyama. Decoupling the class label and the target concept in machine unlearning. *arXiv preprint arXiv:2406.08288*, 2024.

Rui Zhu, Di Tang, Siyuan Tang, XiaoFeng Wang, and Haixu Tang. Selective amnesia: On efficient, high-fidelity and blind suppression of backdoor effects in trojaned machine learning models. In *2023 IEEE Symposium on Security and Privacy (SP)*, pp. 1–19. IEEE, 2023.

Zhiwei Zuo, Zhuo Tang, Kenli Li, and Anwitaman Datta. Machine unlearning through fine-grained model parameters perturbation. *arXiv preprint arXiv:2401.04385*, 2024a.

Zhiwei Zuo, Zhuo Tang, Bin Wang, Kenli Li, and Anwitaman Datta. ecil-mu: Embedding based class incremental learning and machine unlearning. *arXiv preprint arXiv:2401.02457*, 2024b.

A APPENDIX

The appendix is organized as follows.

- In Appendix A.1, we summarize all the notations used in this paper.
- In Appendix A.2, we provide the evaluation metric definitions and formulas used in our experiments.
- In Appendix A.3, we present all the existing evaluation metrics and the frequency they appear.
- In Appendix A.4, we discuss the relationship between retraining and the approximate machine unlearning.
- In Appendix A.5, we provide additional quantitative evidence on point-estimate retraining raised during review.
- In Appendix A.6, we introduce the experiment platform we use in this paper.
- In Appendix A.7 and Appendix A.8, we present the extended experiment results for all the datasets and models.
- In Appendix A.9, we analyze how individual hyper-parameters impact the retraining process.

A.1 NOTATION TABLE

A.2 EVALUATION METRICS FOR MACHINE UNLEARNING

In this section, we introduce several widely used evaluation metrics in evaluating machine unlearning algorithms.

Accuracy is one of the most basic metrics used in MU. Almost all prior work report accuracy on the remain set D_r and test set D_{te}, which infers preserved utility, as well as on the forget set D_f, indicating how many predictions change after unlearning. We focus on a recently proposed aggregate accuracy metric, named the adaptive unlearning score (Cotogni et al., 2023), which we describe as follows.

Adaptive Unlearning Score (AUS) AUS (Cotogni et al., 2023) is a metric accounting for both the test accuracy $\mathcal{A}_{te}$ and the forget accuracy $\mathcal{A}_f$ of the unlearned model:

$$\text{AUS} = \frac{1 - \left(\mathcal{A}_{te}^{Or} - \mathcal{A}_{te}\right)}{1 + \Delta}, \quad \Delta = \begin{cases} |0 - \mathcal{A}_f| & \text{class-wise} \\ |\mathcal{A}_{te} - \mathcal{A}_f| & \text{random} \end{cases}$$

In class-wise setting, Δ in the denominator is the absolute value of the forget set accuracy, suggesting a good class-wise unlearner should have forget set accuracy close to 0. In contrast, Δ is the difference between the $\mathcal{A}_{te}$ and $\mathcal{A}_f$ in the random setting. Δ is small when the forget set accuracy is close to that of the test set, which means that the unlearned model should achieve accuracy similar to unseen data on the forget set after unlearning. In the numerator, the difference between the original and unlearned models' test accuracy acts like a penalty term.

Membership Inference Attack (MIA) Membership inference attack (MIA) (Shokri et al., 2016; Carlini et al., 2022) has been widely adopted to audit the quality of unlearning of MU algorithms and models, which is not always accurately captured by traditional machine learning metrics such as accuracy. While these attacking methods differ in their setups, we identify two major lines of attacks. The first approach, which we denote as MIA-F, allows an attacker to access and separate the outputs of the unlearned model from both the **F**orget set D_f and the test set D_{te} (Carlini et al., 2022; Choi & Na, 2023; Jung et al., 2024; Cotogni et al., 2023; Shah et al., 2023; Chen et al., 2021b; Kurmanji et al., 2023b; Di et al., 2024). Formally, for an unlearned model $\boldsymbol{\theta}_u$, datasets $D_f = \{x_i^f, y_i^f\}_{i=1}^N$ and $D_{te} = \{x_i^{te}, y_i^{te}\}_{i=1}^N$, and a set of model outputs with their membership labels $\mathcal{S} = \{M(x_i; \boldsymbol{\theta}_u), y_i^{mem}\}_{i=1}^{2N}$ (where y^{mem} denotes the membership label $\mathbb{I}[x_i \in D_f]$ ($\mathbb{I}[\cdot]$ is the indicator function) , an attacker a's accuracy is calculated as

Notation	Description
D_{tr}	The **tr**aining set
D_f	The **f**orgetting set
D_{te}	The **te**sting set
D_r	The **r**emaining set and $D_r = D_{tr} \backslash D_f$
$\boldsymbol{\theta}_{tr}$	The parameters of the model trained on D_{tr}
$\boldsymbol{\theta}_r$	The parameters of the model trained on D_r
$\boldsymbol{\theta}_u$	The parameters of the unlearned model
$\|D\|$	The size of the dataset D
$\mathcal{A}^f$	The accuracy on D_f of the unlearned model
$\mathcal{A}^{te}$	The accuracy on D_{te} (TA) of the unlearned model
$\mathcal{A}^{tr}$	The training accuracy or the accuracy on D_{tr} of the unlearned model
$\mathcal{A}^r$	The remaining accuracy (RA) or the accuracy on D_r of the unlearned model
$\mathcal{A}^u$	The unlearn accuracy and $\mathcal{A}^u = 1 - \mathcal{A}^f$
$\mathcal{A}_{te}^{Or}$	$\mathcal{A}^{te}$ of the original model
$\mathcal{A}_{te}^{Re}$	$\mathcal{A}^{te}$ of the retrained model
$\mathcal{A}_{te}^{Un}$	$\mathcal{A}^{te}$ of the unlearned model
A	The learning algorithm
U_e	The exact MU process
U	The unlearning algorithm
$\mathcal{D}$	The example space
$\mathcal{H}$	The hypothesis space of models
Θ	The model space
$\mathcal{R}$	The output region
$\mathbb{P}[\cdot]$	The probability of the event
$\mathcal{S}$	The dataset for training the membership inferencne attackers
M	The forward of a model and the data
x	The input data
y	The target label
$\mathbb{I}$	The indicator function
$\mathbb{D}$	The distance function
L_f	The loss distribution on the forget set
L_{te}	The loss distribution on the test set

Table 2: The notation used in this paper

$$\text{MIA-F}(x, a, \boldsymbol{\theta}_u) = \frac{1}{2N} \sum_{x \in D_f \cup D_{te}} \mathbb{I}[a(M(x; \boldsymbol{\theta}_u)) = y^{mem}].$$

In this context, a prediction of 1 indicates membership, while 0 indicates non-membership. The goal of the attacker is to discern between the outputs from D_f and D_{te}. Inability to do so, reflected by an accuracy near 50%, implies that the model cannot tell the membership difference between the unlearned model's prediction on D_f and D_{te}. In other words, the attacker is unable to tell, based on an unlearned model's output, how forget samples are different from unseen samples, indicating effective unlearning. In practice, the attacker is trained on a portion of $\mathcal{S}$, and the MIA-F score is reported on the rest of the samples.

The second approach to MIA trains an attacker to distinguish between the unlearned model's output from the **R**emaining set, D_r, and the test set, D_{te} (Wu & Harandi, 2024; Jia et al., 2023; Fan et al., 2023; Kodge et al., 2023; Tarun et al., 2023b; Fan et al., 2024; Wu & Harandi, 2025). The attacker then attempts to infer the membership of samples in D_f. We denote this approach by MIA-R. Similarly, for an unlearned model $\boldsymbol{\theta}_u$, datasets $D_r = \{x_i^r, y_i^r\}_{i=1}^N$ and $D_{te} = \{x_i^{te}, y_i^{te}\}_{i=1}^N$, the attacker a is trained on a set of model outputs with their membership labels $\mathcal{S} = \{M(x_i), y_i^{mem}\}_{i=1}^{2N}$ (here $y_i^{mem} = \mathbb{I}[x_i \in D_r]$), and the MIA-R score is calculated as the attacker a's true negative rate

$$\text{MIA-R}(x, a, M) = \frac{1}{|D_f|} \sum_{x \in D_f} \mathbb{I}[a(M(x; \boldsymbol{\theta}_u)) = 0],$$

where $|D_f|$ represents the size of the forget set. Sepcifically, the attacker quantifies the percentage of the samples in the forget set being classified as non-membership. A higher MIA-R score in this case means successfully "misleading" the attacker. Therefore, a robust unlearning algorithm is expected to yield a high true negative rate, predominantly classifying forget set samples as non-members (Jia et al., 2023). Higher scores suggest unsuccessful membership inference, affirming the algorithm's efficacy in unlearning.

In summary, the MIA-F score represents the attacker's confidence in discerning the unlearned model's output on the D_f and D_{te} while the MIA-R score means the percentage of non-membership correctly predicted by the attacker.

Distance Functions Following previous work, we employ two distance functions: L2 distance (Poppi et al., 2023; Yoon et al., 2022; Chundawat et al., 2023a; Tarun et al., 2023a; Thudi et al., 2022a; Izzo et al., 2021) and Wasserstein distance (Wu & Harandi, 2024). We use L2 distance to measure weight-space shifts between retrained and original pre-trained models, and 1-Wasserstein distance to compare loss distributions on D_f and D_{te}.

L2 distance is usually used to compare the weight space distance between the unlearned model and the retrained model, and a smaller distance indicates being closer to the retrained model, and thus better unlearning (Thudi et al., 2022a; Tarun et al., 2023a). Formally, given two model weight vector $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$, the L2 distance between the two weight vectors is defined as follows:

$$\mathbb{D}_{\text{L2}}(\boldsymbol{\theta}, \boldsymbol{\theta'}) = \sqrt{\sum_i (\boldsymbol{\theta}_i - \boldsymbol{\theta}'_i)^2}$$

where, $\boldsymbol{\theta}_i$ denotes the i-th parameter in the weight $\boldsymbol{\theta}$. Since we only employ retrained models, we measure the L2 distance of the retrained model relative to the pre-trained model. While a larger L2 distance does not necessarily infer better unlearning (Thudi et al., 2022b), our goal is to show that, depending on the retraining setup, retrained models can vary drastically in the weight space. We present evidence to this claim in Section 4.2.

We also consider p-Wasserstein distance (p-WD) (Arjovsky et al., 2017) for sample-wise loss computed from an unlearned model's outputs on D_f and D_{te}. For 1-Wasserstein distance computed on empirical discrete distributions (Wu & Harandi, 2024), it is defined as the L1 norm of the ordered samples in the two distributions. In our formulation, we consider two loss distributions L_f and L_{te}, from the forget set and the test set, respectively. We then have:

$$\mathbb{D}_{1-\text{WD}}(L_f, L_{te}) = \frac{1}{n} \sum_i |l_f^i - l_{te}^i|$$

where, l_f^i and l_{te}^i represent the loss computed on the i-th sample in the forget set and test set. A small 1-WD indicates that the loss distribution on the forget set is similar to that on the test set.

Average Gap Accuracy, MIA scores, and distance function all evaluate an unlearned model's utility and unlearning efficacy in some dimension. Fan et al. (2024) proposed to use an aggregate metric, namely the average gap, to reflect both model utility and effectiveness of unlearning (Fan et al., 2023). Unlike AUS, which incorporates accuracy on different sets, average gap uses the retrained model as a reference, and is calculated as the averaged absolute value of the gaps between a retrained model and an unlearned model, in remain accuracy, test accuracy, the complement of the forget accuracy, and MIA score.

If we use the superscript Re and Un to denote the metric calculated on the retrained and unlearned models, respective, the average gap is computed as:

$$\begin{aligned}\text{AvgGap} = \frac{1}{4}(&\left|\mathcal{A}_r^{Re} - \mathcal{A}_r^{Un}\right| + \left|\mathcal{A}_f^{Un} - \mathcal{A}_f^{Re}\right| \\ &+ \left|\mathcal{A}_{te}^{Re} - \mathcal{A}_{te}^{Un}\right| + \left|\text{MIA}^{Re} - \text{MIA}^{Un}\right|)\end{aligned} \tag{1}$$

Here, Re and Un refer to the retrained and unlearned models. r, f, te represent the remain set, forget set, and test set, respectively. $\mathcal{A}_f$ is the complement of the forget set accuracy (1 - forget set accuracy). Average gap has already been employed in Fan et al. (2023) and Wu & Harandi (2024).

A.3 Machine Unlearning Evaluation Metrics

In this section, we review existing evaluation metrics for machine unlearning algorithms in classification tasks. We then discuss the role of retraining, an exact unlearning method, and how it is used as a evaluation standard to benchmark other approximate unlearning methods.

Accuracies on the remain set, forget set, and test set are commonly reported in the literature of machine unlearning (Nguyen et al., 2022; Shaik et al., 2023; Xu et al., 2023). $\mathcal{A}_{te}$ quantifies the utility of the unlearned model, while forget set accuracy indicates the extent of change in predictions after unlearning. To account for multiple measures, aggregate accuracy metrics have been utilized. Cotogni et al. (2023) introduced an adaptive score that captures the trade-off between test accuracy and forget accuracy, and also considers the varying difficulty between classwise and random (homogeneous) forgetting (Jia et al., 2023; Fan et al., 2023). Choi & Na (2023) and Jung et al. (2024) linearly combine test accuracy with MIA accuracy to quantify both utility and the quality of forgetting, applying a coefficient to prioritize between the two scores. Similarly, Shibata et al. (2021) and Kim et al. (2023a) calculate a normalized score based on remain and forget accuracy. Liu et al. (2022) defines the unlearning score as a ratio of remain accuracy to forget accuracy.

Membership inference attack The use of membership inference attacks (MIAs) in evaluation is a recurring theme in many studies. These approaches vary in three key aspects: 1) data sources and 2) information extracted from the examined model outputs. A common method is allowing the attacker to access the outputs of the target model from the remain set and test set. Then the attacker tries infer the membership of the forget set samples (Jia et al., 2023; Fan et al., 2023; Kodge et al., 2023). In this setting, if a forget sample can be correctly inferred as a member, the attack if successful. An alternative method gives the attacker model access to outputs from both the forget set and an unseen set, aiming to predict whether a data point was used in the training set (Choi & Na, 2023; Shah et al., 2023; Cotogni et al., 2023; Jung et al., 2024). Assuming white-box access to the target model, one could exploit multiple features from the target model weights or output to perform the attack. Popular ones include prediction probabilities (Jia et al., 2023; Fan et al., 2023; Cotogni et al., 2023), output probability entropy (Golatkar et al., 2021; 2020b), and losses (Choi & Na, 2023; Shah et al., 2023; Jung et al., 2024). Shi et al. (2023) additionally incorporates gradients and activations from the final layer in their attacker model training process. We provide a more detailed formulation of MIA methods used in our experiments in Appendix A.2.

Distance functions Based on the definition of machine unlearning (Nguyen et al., 2022) on the output distribution and the weight distribution, distance functions are used to measure the similarity between the retrained model and the unlearned model in both the output space and the weight space. These include include L1 or L2 distance (Golatkar et al., 2021; Poppi et al., 2023; Chundawat et al., 2023a; Tarun et al., 2023a; Thudi et al., 2022a; Izzo et al., 2021), 1-Wasserstein distance (Wu & Harandi, 2024), and Jensen-Shannon divergence (Poppi et al., 2023). Another metric known as the zero retrain forgetting score (ZRF) is proposed as a retrain-free metric to evaluate unlearned models (Poppi et al., 2023; Chundawat et al., 2023a). ZRF measures the the randomness in forget set prediction by comparing the unlearned model's output with a randomly initialized model's output, with a larger score indicating more random predictions on the forget set.

A.4 Evaluate an approximate MU using Retrained Model

An approximate MU with a closer performance to the retrained model is regarded a better MU method because retrained model is treated as the exact MU. **It should be noted that although the retrained**

Metrics	Papers
Test accuracy	(Jung et al., 2024; Wu & Harandi, 2024; Zuo et al., 2024a;b; Kim et al., 2023a; Li et al., 2023; Cotogni et al., 2023; Kodge et al., 2023; Shah et al., 2023; Abbasi et al., 2023; Shi et al., 2023; Choi & Na, 2023; Chen & Yang, 2023; Fan et al., 2023; Liu et al., 2022; Foster et al., 2023; Li & Ghosh, 2023; Jia et al., 2023; Poppi et al., 2023; Chen et al., 2023; Kurmanji et al., 2023b; Cha et al., 2023; Yoon et al., 2022; Chundawat et al., 2023b; Chen et al., 2021a; Tarun et al., 2023a; Chen et al., 2021b; Warnecke et al., 2021; He et al., 2021; Liu et al., 2020; Goel et al., 2022; Zhang et al., 2023; Zhu et al., 2023; Chundawat et al., 2023a; Thudi et al., 2022a; Graves et al., 2021; Golatkar et al., 2021; 2020a;b; Ebrahimpour-Boroojeny et al.; Fan et al., 2024; Huang et al., 2024a;b; Wu & Harandi, 2024; Patel & Qiu, 2025; Wu & Harandi, 2025; Lin et al., 2024; Zhu et al., 2024; Yang et al., 2025)
Remaim accuracy	(Poppi et al., 2023; Wu & Harandi, 2024; Zuo et al., 2024b; Kim et al., 2023a; Li et al., 2023; Cotogni et al., 2023; Kodge et al., 2023; Shah et al., 2023; Abbasi et al., 2023; Shi et al., 2023; Chen & Yang, 2023; Fan et al., 2023; Liu et al., 2022; Zhang et al., 2023; Foster et al., 2023; Jia et al., 2023; Chen et al., 2023; Kurmanji et al., 2023b; Cha et al., 2023; Yoon et al., 2022; Chundawat et al., 2023a;b; Tarun et al., 2023a; Graves et al., 2021; Golatkar et al., 2021; 2020a;b; Ebrahimpour-Boroojeny et al.; Fan et al., 2024; Huang et al., 2024a;b; Wu & Harandi, 2024; Patel & Qiu, 2025; Wu & Harandi, 2025; Lin et al., 2024; Zhu et al., 2024; Yang et al., 2025)
Forget accuracy	(Wu & Harandi, 2024; Zuo et al., 2024b; Kim et al., 2023a; Cotogni et al., 2023; Kodge et al., 2023; Shah et al., 2023; Abbasi et al., 2023; Shi et al., 2023; Chen & Yang, 2023; Fan et al., 2023; Liu et al., 2022; Zhang et al., 2023; Foster et al., 2023; Li & Ghosh, 2023; Jia et al., 2023; Poppi et al., 2023; Chen et al., 2023; Cha et al., 2023; Yoon et al., 2022; Chundawat et al., 2023a;b; Tarun et al., 2023a; Graves et al., 2021; Liu et al., 2020; Fan et al., 2024; Huang et al., 2024a;b; Wu & Harandi, 2024; Patel & Qiu, 2025; Wu & Harandi, 2025; Lin et al., 2024; Zhu et al., 2024; Yang et al., 2025)
Adaptive unlearning score	(Cotogni et al., 2023)
MIA-R	(Wu & Harandi, 2024; Jia et al., 2023; Fan et al., 2023; Kodge et al., 2023; Tarun et al., 2023b; Shi et al., 2023; Golatkar et al., 2021; Fan et al., 2024; Huang et al., 2024a;b; Wu & Harandi, 2024; Patel & Qiu, 2025; Wu & Harandi, 2025; Lin et al., 2024; Zhu et al., 2024; Yang et al., 2025)
MIA-F	(Choi & Na, 2023; Jung et al., 2024; Cotogni et al., 2023; Shah et al., 2023; Chen et al., 2021b; Kurmanji et al., 2023b; Ebrahimpour-Boroojeny et al.)
L2 distance (output or weight)	(Poppi et al., 2023; Yoon et al., 2022; Chundawat et al., 2023a; Tarun et al., 2023a; Thudi et al., 2022a)
1-Wasserstein distance (output)	(Wu & Harandi, 2024)
KL divergence (weight)	(Thudi et al., 2022a)
Jensen-Shannon divergence (output)	(Zuo et al., 2024a; Poppi et al., 2023; Chundawat et al., 2023a)
Average gap	(Jia et al., 2023; Fan et al., 2023; Wu & Harandi, 2024;?; Patel & Qiu, 2025; Wu & Harandi, 2025; Lin et al., 2024; Zhu et al., 2024; Yang et al., 2025)
Relearn time	(Tarun et al., 2023a; Li et al., 2023; Golatkar et al., 2021; 2020a;b; Li et al., 2023)
Anamnesis index	(Abbasi et al., 2023; Chundawat et al., 2023b)
Zero retrain forgetting score	(Poppi et al., 2023; Chundawat et al., 2023a)
Interclass confusion	(Kurmanji et al., 2023b; Goel et al., 2022)
Gradient residual norm	(Warnecke et al., 2021; Guo et al., 2019)
Model inversion attack	(Graves et al., 2021)
Epistemic uncertainty	(Becker & Liebig, 2022)

Table 3: The existing MU evaluation metrics and their papers

model is treated as the exact MU, retraining method itself is one of the MU methods. Thus, all the evaluation metrics used to evaluate the approximate MU can be used to evaluate the retrained model. The metrics of the retrained model will be regarded as the evaluation standard of the other approximate MU methods. If the metrics of the standard is not stable, it is hard to evaluate the other approximate objectively, which is the meaning of this paper.

A.5 Additional Quantitative Evidence on Point-Estimate Retraining

We include additional numerical evidence that illustrates why a single retrained checkpoint can be a fragile reference even when model utility is held nearly fixed. On ResNet-18/CIFAR-10 with random forgetting, we identify retrained models whose test accuracy lies in the narrow band $[0.940, 0.950]$. Despite this small utility range, MIA-R varies substantially, as shown in Table 4.

Retraining configuration	Test Acc.	MIA-R
lr = 0.3, wd = 1e-4, cosine	0.9475	0.383
lr = 0.03, wd = 3e-4, cosine	0.9449	0.252
lr = 0.3, wd = 5e-4, step	0.9415	0.371
lr = 0.03, wd = 3e-4, step	0.9428	0.258

Table 4: Utility-matched retraining examples on ResNet-18/CIFAR-10 with random forgetting. Although test accuracy remains within a 1% band, MIA-R changes by 0.149 across these configurations.

The same issue appears even when all retraining hyperparameters are fixed to the original pre-training recipe and only the random seed changes. In Table 5, test accuracy is nearly unchanged across seeds, while MIA-R still exhibits a much larger spread.

Metric	Seed 1	Seed 2	Seed 3	Std.
Test Acc.	0.9519	0.9515	0.9510	0.0004
MIA-R	0.321	0.449	0.395	0.052

Table 5: Seed-only retraining variability under the original ResNet-18/CIFAR-10 recipe (lr = 0.1, wd = 5e-4, cosine schedule). Even without hyperparameter drift, MIA-R varies substantially while test accuracy is stable.

Table 6 further compares the original recipe with a nearby weight-decay drift. MIA-F remains stable under both references, whereas MIA-R changes substantially under the original recipe and tightens under the neighboring drift. This supports our recommendation to treat MIA-F as a more robust default signal and MIA-R as a diagnostic metric.

Retraining setting	Test Acc. (seed 1/2/3)	MIA-R (seed 1/2/3)	MIA-F (seed 1/2/3)
Original recipe, wd = 5e-4	0.9519 / 0.9515 / 0.9510	0.3212 / 0.4488 / 0.3952	0.5069 / 0.5028 / 0.5039
Nearby drift, wd = 3e-4	0.9509 / 0.9500 / 0.9491	0.3464 / 0.3594 / 0.3648	0.5062 / 0.5054 / 0.5012

Table 6: Original-recipe and nearby-drift retraining results on ResNet-18/CIFAR-10 random forgetting. MIA-F remains stable across both references, while MIA-R is sensitive to the selected retrained reference.

We also test whether the same point-estimate concern can appear beyond small image classifiers. On MUSE, a text-generation unlearning benchmark based on Llama-2-7B (Shi et al., 2024), Privacy Leakage is retrain-dependent. As shown in Table 7, three retrained seeds produce values spanning nearly 0.6 and changing sign, which means that a single retrained reference can change the qualitative interpretation of the metric. For the unlearning method SimNPO evaluated against the same three retrained references, the corresponding Privacy Leakage values are -0.17, +0.22, and +0.43.

Retrained reference	Privacy Leakage
retrain seed 1	0.3292
retrain seed 2	-0.0621
retrain seed 3	-0.2671

Table 7: Additional MUSE evidence showing that a retrain-dependent privacy metric can change substantially across retrained seeds.

A.6 Experimental Platform

We conduct most experiments on a node with 8 NVIDIA RTX 4090 GPUs. All experiments take about two weeks to complete. The Pytorch version we use is 2.4.1 on a Ubuntu 22.04 system.

A.7 Additional Evidence on Other Datasets and Model Architectures

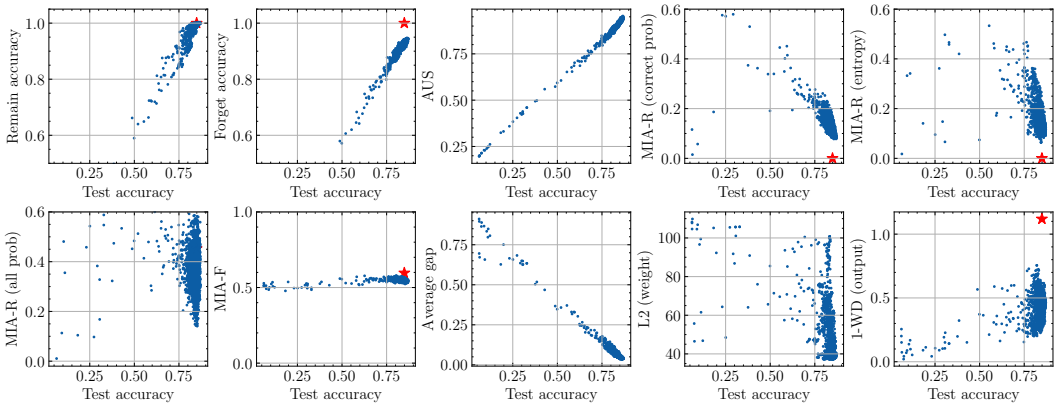


Figure 7: The relationship between test accuracy and 10 metrics used to evaluate machine unlearning models, including accuracy, MIA, distance, and aggregate metrics, in the context of **random** forgetting using BERT on 20 Newsgroups. While the fluctuation in MIA-R (correct prob) is not as strong as in image classification settings in Figure 2, we still observe instability in other MIA-R variants and distance function metrics. This figure is used as supporting evidence for Figure 2.

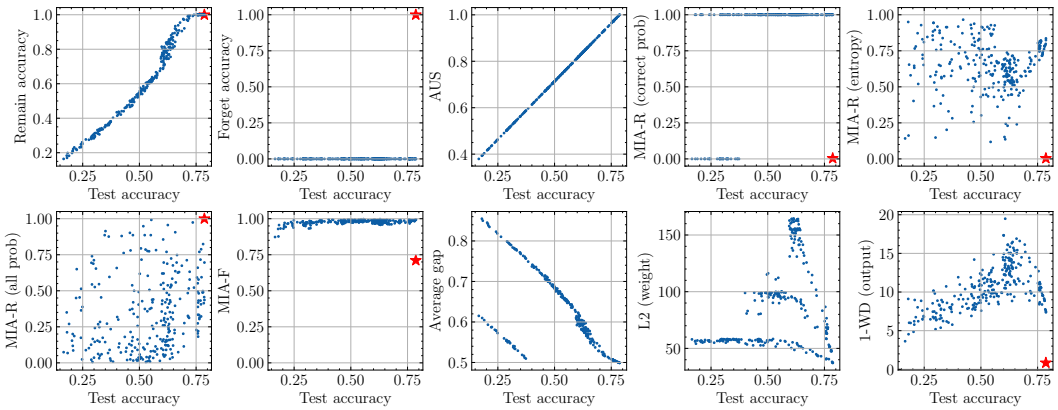
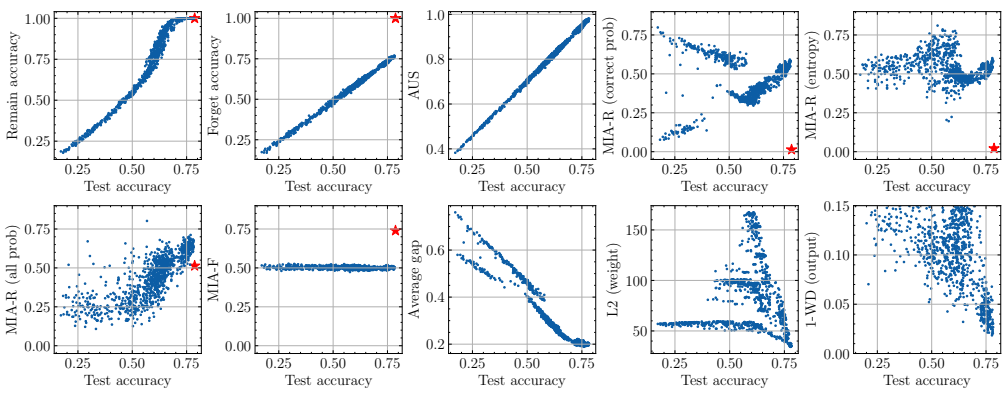
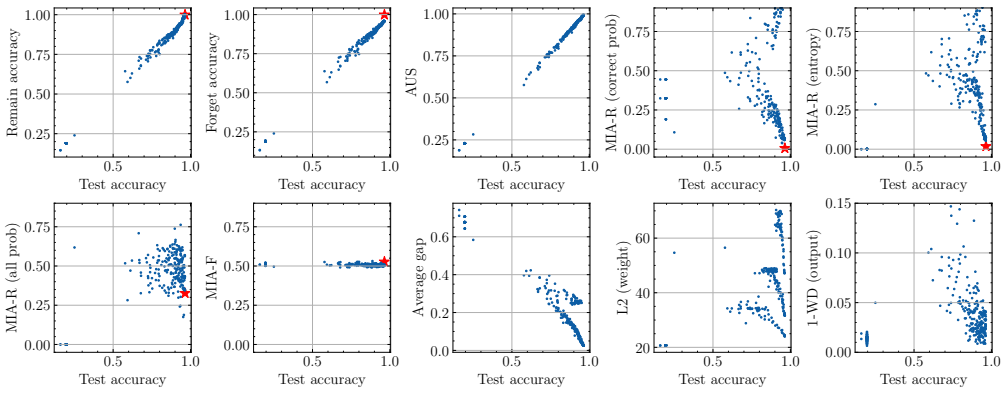


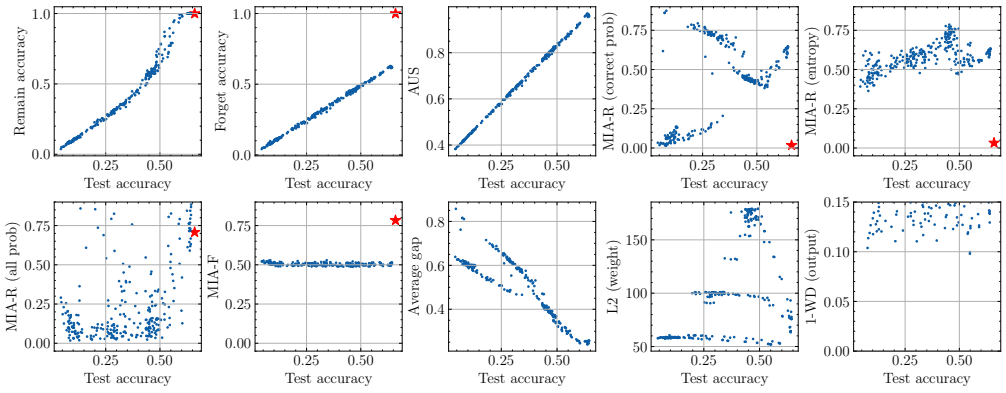
Figure 8: All metrics for classwise forgetting on CIFAR-100 with ResNet-18, supporting Figure 1.



(a) CIFAR-100

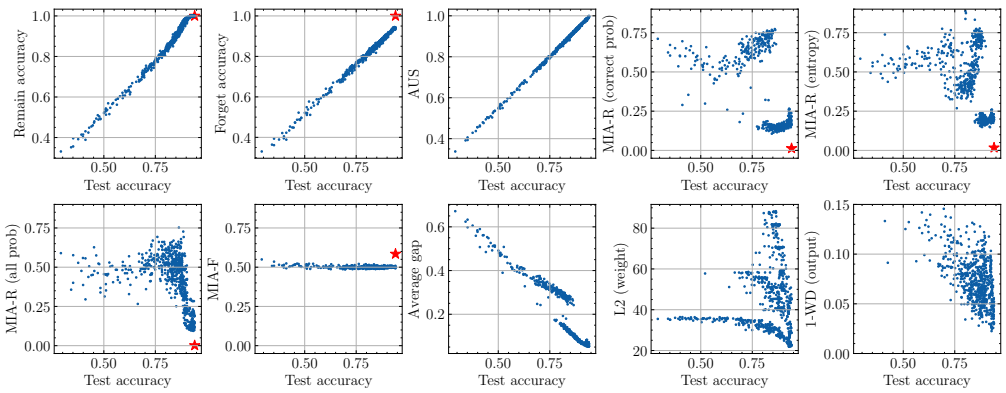


(b) SVHN

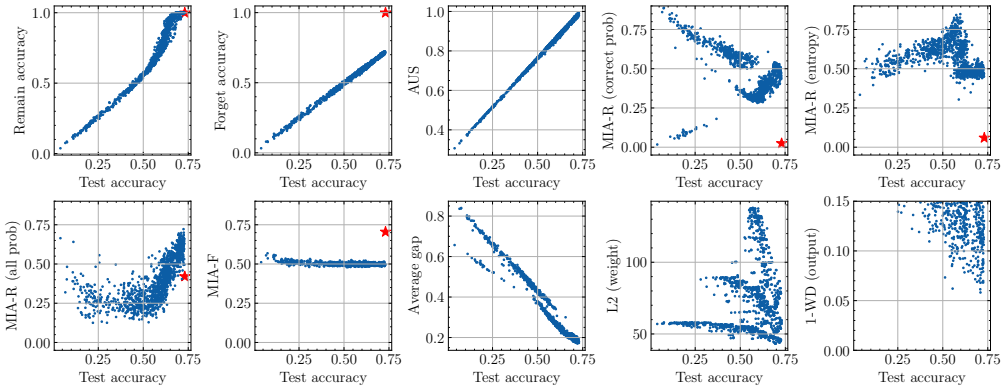


(c) Tiny-ImageNet

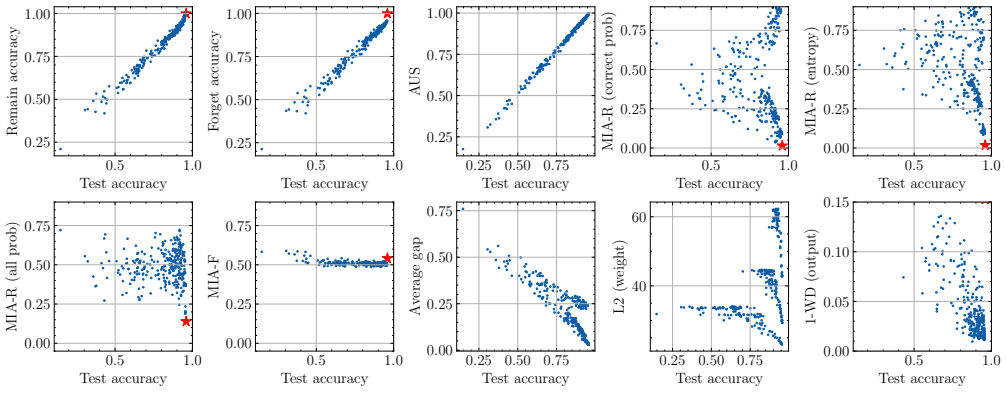
Figure 9: All metrics for random forgetting on ResNet-18 trained on CIFAR-100, SVHN, and Tiny-ImageNet, supporting Figure 2.



(a) CIFAR-10

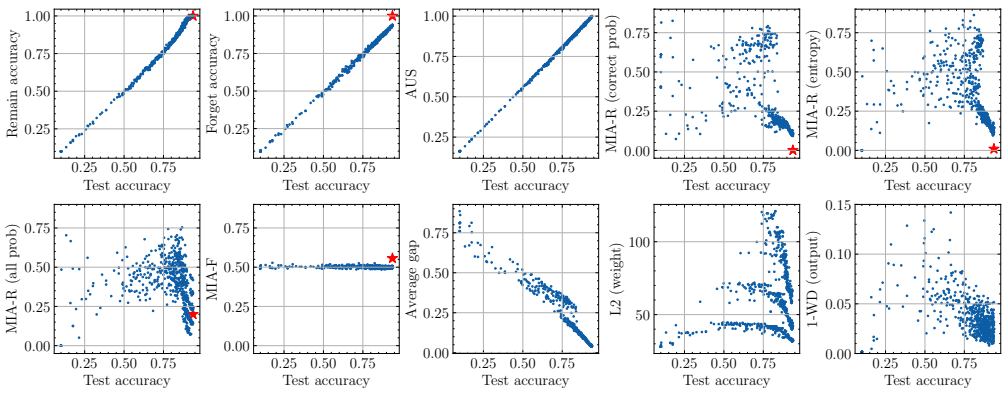


(b) CIFAR-100

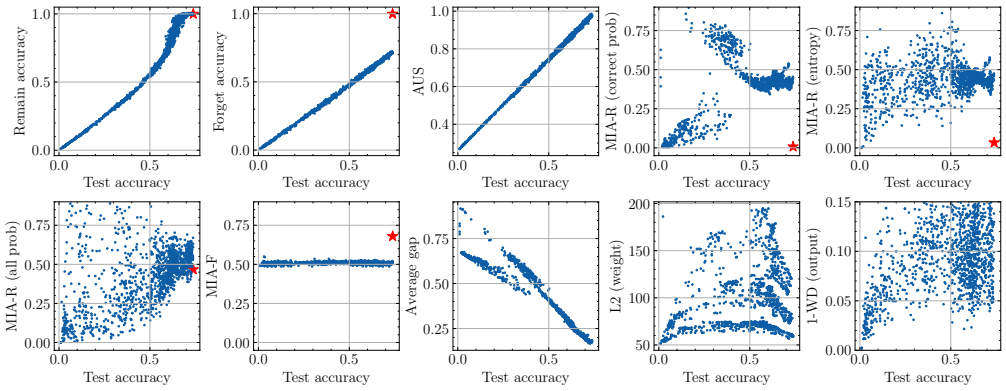


(c) SVHN

Figure 10: All metrics for random forgetting on ALLCNN trained on CIFAR-10, CIFAR-100, and SVHN, supporting Figure 2.



(a) CIFAR-10



(b) CIFAR-100

Figure 11: All metrics for random forgetting on VGG-16 trained on CIFAR-10 and CIFAR-100, supporting Figure 2.

A.8 ADDITIONAL EVIDENCE ON ALL DATASETS AND MODEL ARCHITECTURES

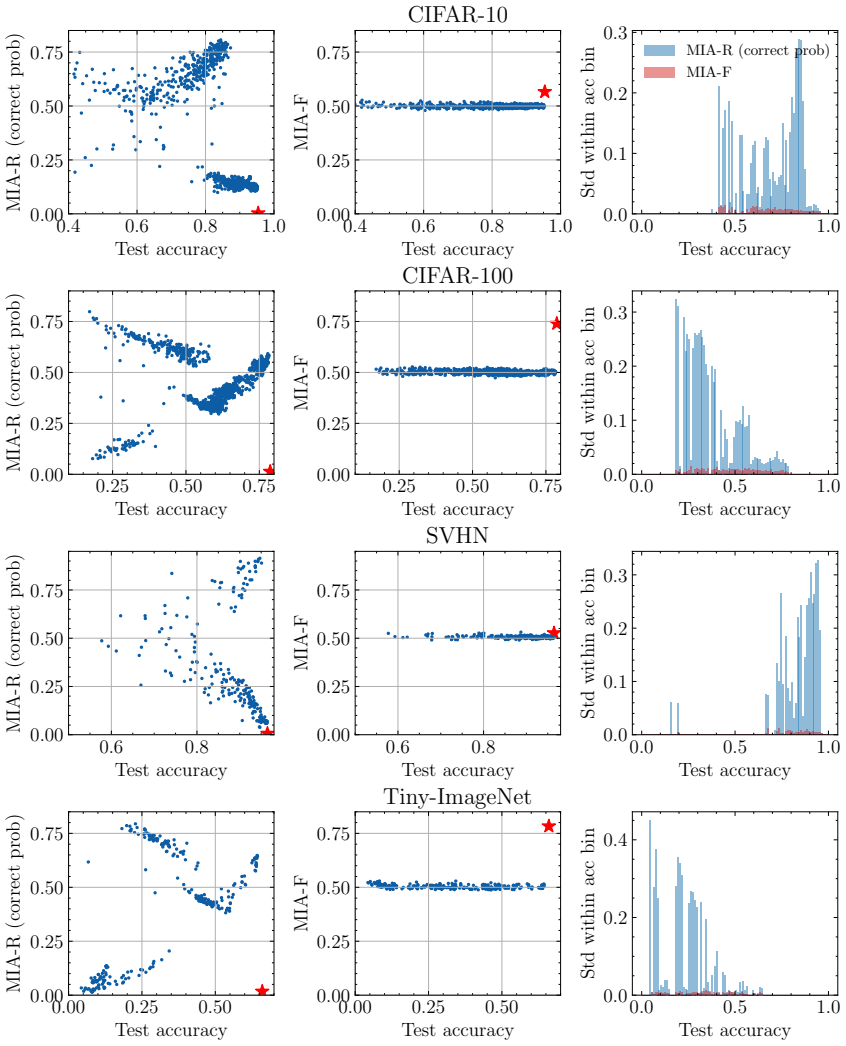


Figure 12: MIA-R (correct prob) versus MIA-F of retrained ResNet-18 on CIFAR-10, CIFAR-100, SVHN, and Tiny-ImageNet. This figure is used as supporting evidence for Figure 4.

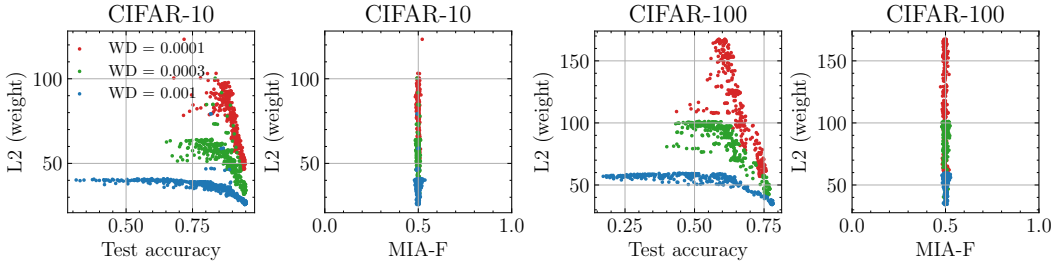


Figure 13: L2 weight distance between retrained and original pre-trained ResNet-18 on CIFAR-10, CIFAR-100, SVHN, and Tiny-ImageNet. This figure is used as supporting evidence for Figure 3.

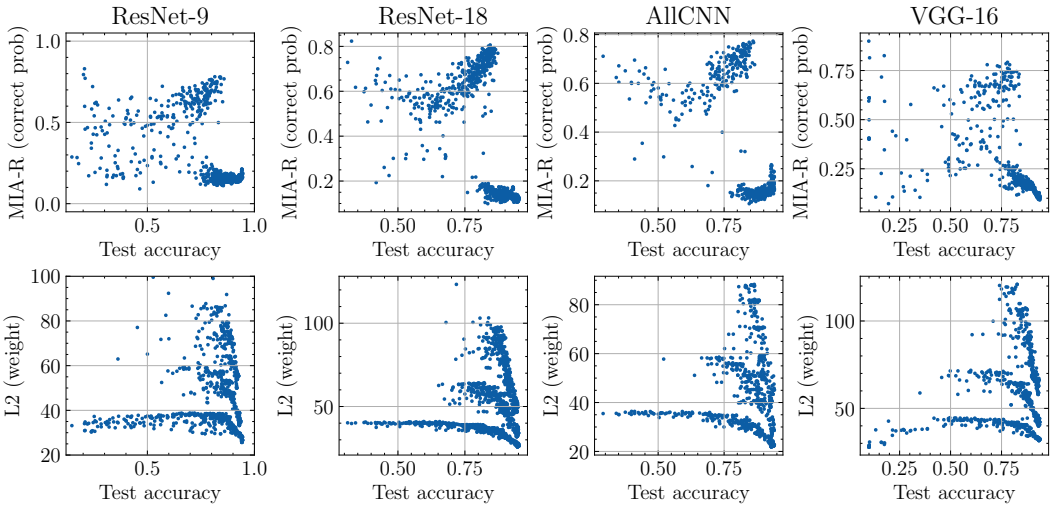
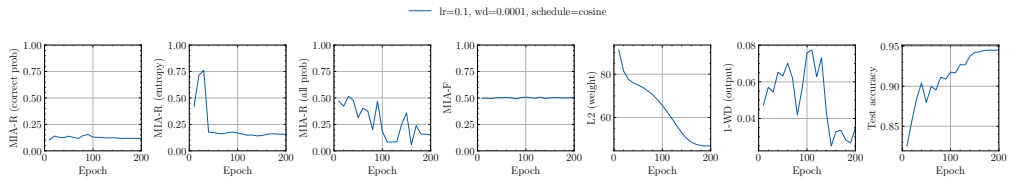


Figure 14: The same instability in both MIA-RT (correct prob) and L2 weight distance hold across all three model architectures, including ResNet-9/18, ALLCNN, and VGG-16, on CIFAR-10.

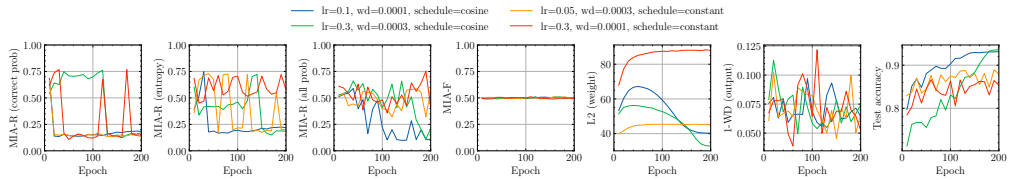
A.9 Analysis on how individual hyper-parameters impact retraining

In this section, we present visualizations of how unlearning metrics evolve across retraining epochs. Each subsequent subsection focuses on the impact of altering a single hyperparameter. We note that instabilities are more prevalent in the initial epochs and are associated with higher learning rates, greater weight decay, and a constant learning rate schedule.

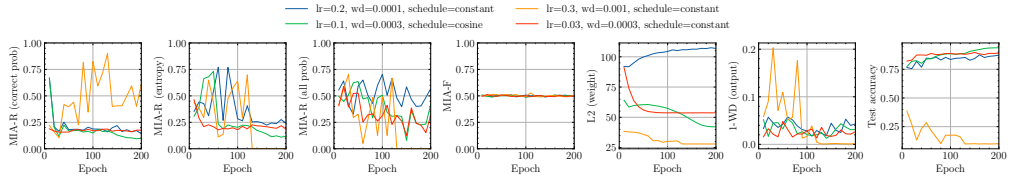
A.9.1 Epochs



(a) ResNet-18

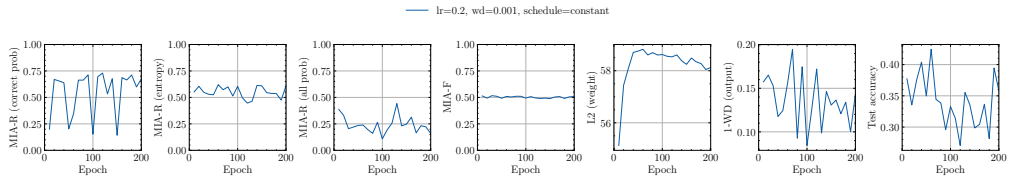


(b) ALLCNN

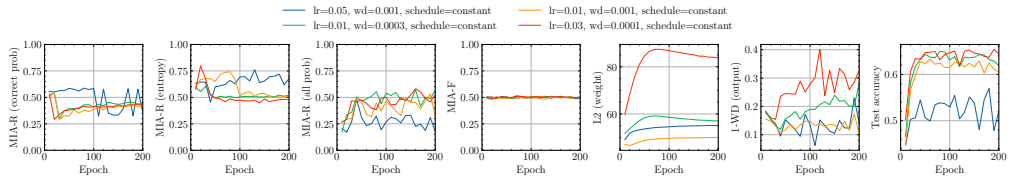


(c) VGG-16

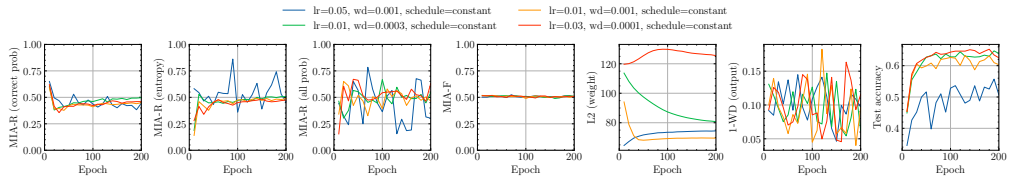
Figure 15: Change in seven machine unlearning metrics as the number of epochs increases, using ResNet-18, ALLCNN, and VGG-16 retrained on D_r of CIFAR-10. For each model, we show four different training trajectories based on the hyperparameter range considered.



(a) ResNet-18

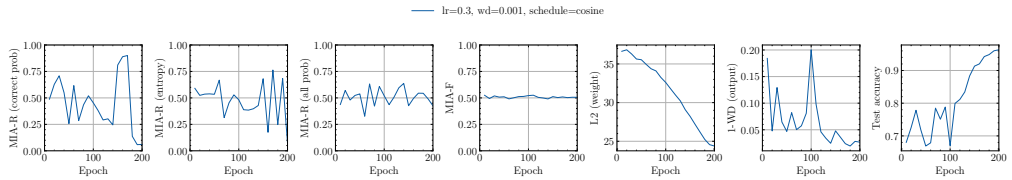


(b) ALLCNN

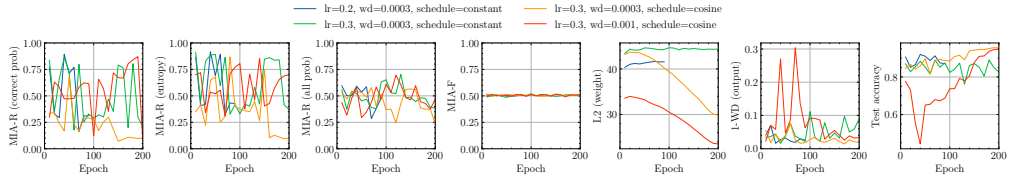


(c) VGG-16

Figure 16: Change in seven machine unlearning metrics as the number of epochs increases, using ResNet-18, ALLCNN, and VGG-16 retrained on D_r of CIFAR-100. For each model, we show four different training trajectories based on the hyperparameter range considered.

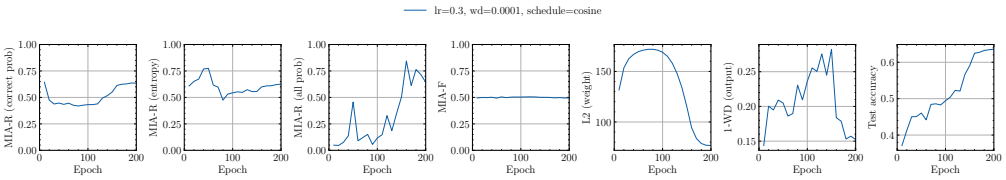


(a) ResNet-18



(b) ALLCNN

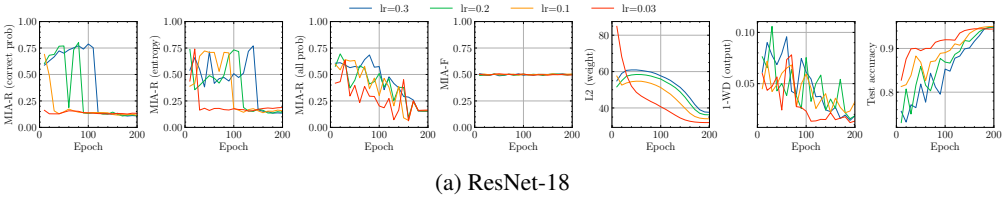
Figure 17: Change in seven machine unlearning metrics as the number of epochs increases, using ResNet-18 and ALLCNN retrained on D_r of SVHN. For each model, we show four different training trajectories based on the hyperparameter range considered.



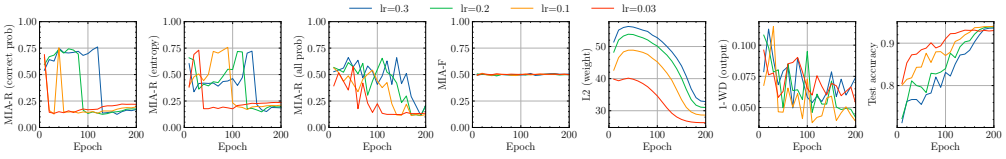
(a) ResNet-18

Figure 18: Change in seven machine unlearning metrics as the number of epochs increases, using ResNet-18 retrained on D_r of Tiny-ImageNet. For each model, we show four different training trajectories based on the hyperparameter range considered.

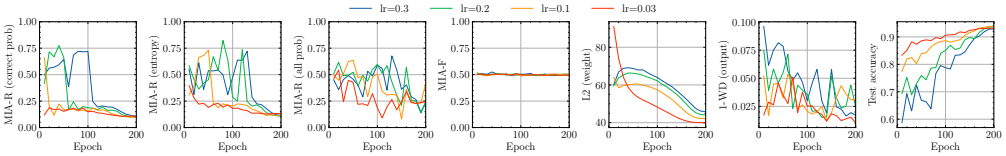
A.9.2 LEARNING RATE



(a) ResNet-18

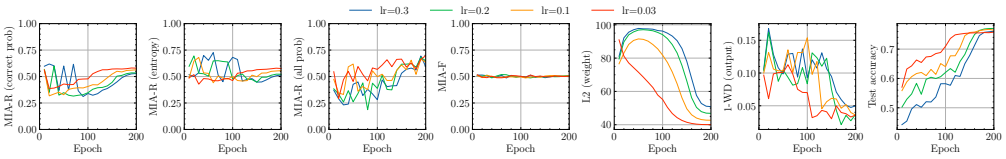


(b) ALLCNN

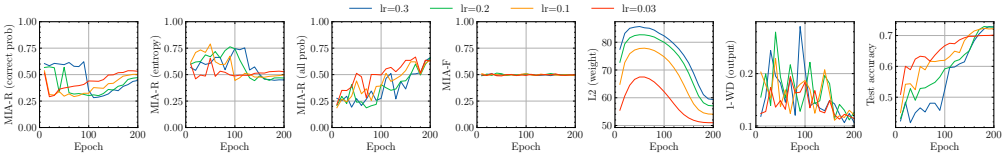


(c) VGG-16

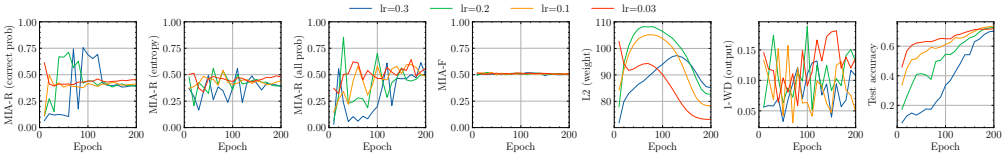
Figure 19: Change in seven machine unlearning metrics on the axis of the number of epochs, by varying only the learning rate and keeping other hyperparameters the same (weight decay = 0.0003 with cosine annealing learning rate schedule), using ResNet-18, ALLCNN, and VGG-16 retrained on D_r of CIFAR-10.



(a) ResNet-18



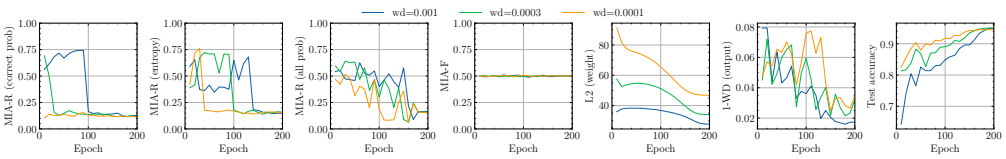
(b) ALLCNN



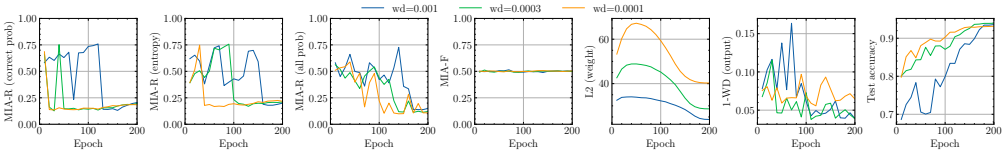
(c) VGG-16

Figure 20: Change in seven machine unlearning metrics on the axis of the number of epochs, by varying only the learning rate and keeping other hyperparameters the same (weight decay = 0.0003 with cosine annealing learning rate schedule), using ResNet-18, ALLCNN, and VGG-16 retrained on D_r of CIFAR-100.

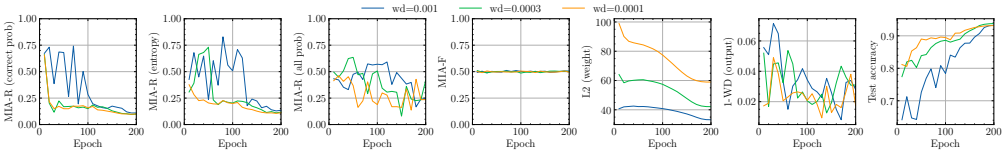
A.9.3 WEIGHT DECAY



(a) ResNet-18

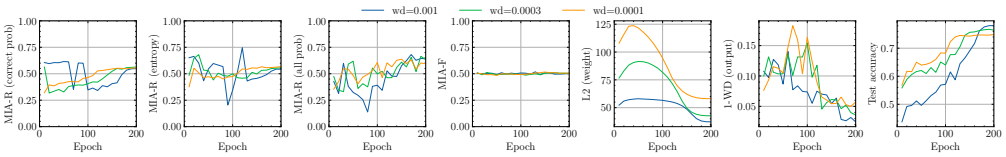


(b) ALLCNN

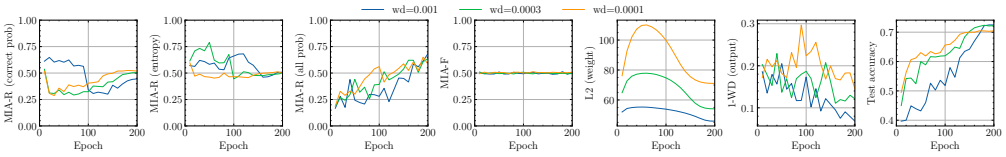


(c) VGG-16

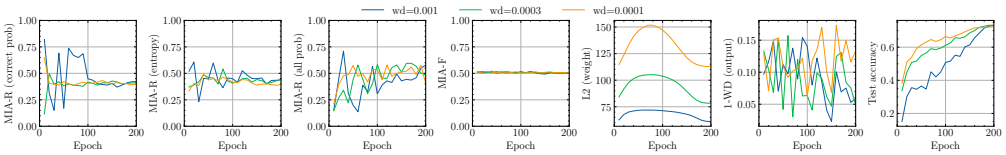
Figure 21: Change in seven machine unlearning metrics on the axis of the number of epochs, by varying only the weight decay and keeping other hyperparameters the same (learning rate = 0.1 with cosine annealing learning rate schedule), using ResNet-18, ALLCNN, and VGG-16 retrained on D_r of CIFAR-10.



(a) ResNet-18



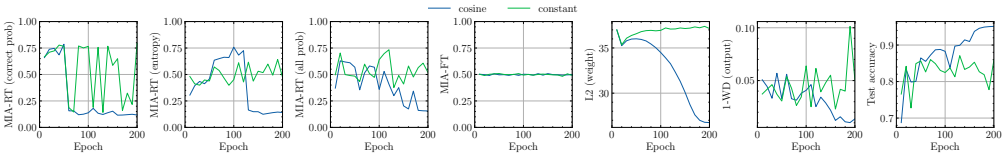
(b) ALLCNN



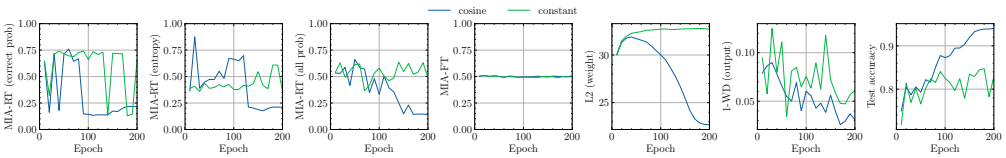
(c) VGG-16

Figure 22: Change in seven machine unlearning metrics on the axis of the number of epochs, by varying only the weight decay and keeping other hyperparameters the same (learning rate = 0.1 with cosine annealing learning rate schedule), using ResNet-18, ALLCNN, and VGG-16 retrained on D_r of CIFAR-100.

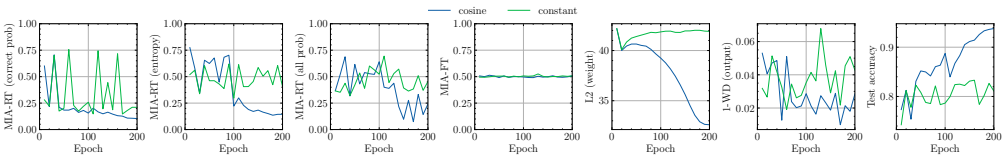
A.9.4 LEARNING RATE SCHEDULER



(a) ResNet-18

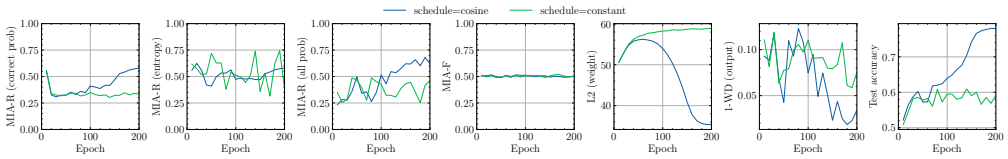


(b) ALLCNN

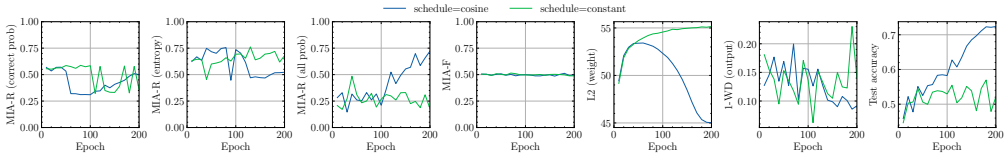


(c) VGG-16

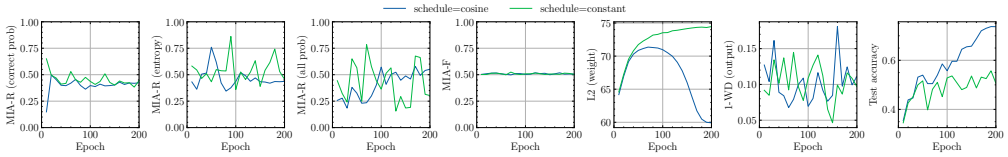
Figure 23: Change in seven machine unlearning metrics on the axis of the number of epochs, by varying only the learning rate schedule and keeping other hyperparameters the same (learning rate = 0.1 and weight decay = 0.0003), using ResNet-18, ALLCNN, and VGG-16 retrained on D_r of CIFAR-10.



(a) ResNet-18



(b) ALLCNN



(c) VGG-16

Figure 24: Change in seven machine unlearning metrics on the axis of the number of epochs, by varying only the learning rate schedule and keeping other hyperparameters the same (learning rate = 0.1 and weight decay = 0.0003), using ResNet-18, ALLCNN, and VGG-16 retrained on D_r of CIFAR-100.