Is Memorization *Actually* Necessary for Generalization?

Hadi Abdullah
Visa Research
habdulla@visa.com

Abstract

Deep learning models are known for their ability to memorize training data. While memorization is often linked to risks such as privacy leakage and poor robustness, a highly influential claim by Feldman and Zhang [12] argues that memorization is actually *necessary* for generalization. Their conclusion is based on the observation that removing points with high memorization scores reduces test accuracy. Upon closer inspection of their work, we uncover four critical flaws in the underlying methodology: **(1) sampling bias** in their approximation algorithm inflates memorization scores; **(2) high false positive rate** in their definition of memorization, leads to misclassification of non-memorized points as memorized; **(3) unprincipled thresholding** that results in an ill-posed problem; and **(4) data leakage** skews the test accuracy results. To address these limitations, we introduce a modifications for correctly identifying and evaluating memorization, including higher sampling rates, modifying the original memorization definition to reduce the false positive rates, proposing a method to identify a principled score threshold, and employing test datasets especially designed to avoid data leakage. Having accounted for these errors, our results show that, in contradiction to the original work, removing truly memorized points does not cause a drop in accuracy, and in most cases, improves test performance. These findings call into question the necessity of memorization in deep learning and highlight the importance of mitigating its risks.

1 Introduction

Deep learning models possess a remarkable ability to *memorize* data. They can fit arbitrary inputs, even when they lie far outside the natural data distribution. For instance, image classifiers can be trained to associate images of Gaussian noise with the label cat [1, 22, 30]. Similarly, LLMs can fit random strings (such as addresses or even secret keys) and regurgitate them when properly prompted [5]. This phenomenon is only possible because of the model's capacity to memorize training data.

At the heart of current literature lies a key tension. On one hand, a substantial body of work has documented the downsides of memorization. They highlight its role in leakage of private data, model stealing attacks, increased model bias, poor robustness to distribution shifts, and susceptibility to adversarial examples [3, 4, 6–8, 13, 16, 19–21, 25, 29]. On the other hand, a highly cited result by Feldman and Zhang [12] claims that memorization is not just incidental but in fact *necessary* for achieving strong generalization in deep networks. This raises an important question: is memorization a necessary evil?

Intuitively, memorization and generalization appear to be at odds. Generalization relies on uncovering patterns that apply to unseen data, while memorization involves storing specific training examples, offering no obvious utility for unseen inputs. However, Feldman and Zhang [12] argue

otherwise. They claim that removing data points with high memorization scores leads to a measurable drop in test accuracy, thereby suggesting that memorization supports generalization.

To make this argument, the authors introduce an approximation algorithm for estimating how much a point is memorized. They then show that removing the most memorized points from the training set decreases test performance. However, a closer inspection reveals several critical flaws in this approach—issues that undermine the core conclusions of the paper:

1. **Sampling Bias:** The approximation algorithm proposed by Feldman [11] introduces sampling bias, resulting in a systematic overestimation of memorization scores (error as high as 10%). This leads to non-memorized points being misclassified as memorized. (Section 3.1)
2. **High False Positive Rate:** The memorization scoring definition is never evaluated against a baseline. We show that it suffers from a high false positive rate (as high as 60% at 95% True Positive Rate), failing to reliably distinguish truly memorized points from naturally learnable ones. (Section 3.2)
3. **Unprincipled Thresholding:** No clear statistical criterion is given for choosing a threshold to distinguish memorized from non-memorized points i.e., the problem is ill-posed. Since the conclusions change based on this arbitrary threshold, the findings lack robustness. (Section 3.3)
4. **Data Leakage:** Some points labeled as highly memorized have near-duplicates in the test set. Thus, the observed test accuracy drop upon removing these points may stem from test-train overlap rather than a causal role of memorization in generalization. (Section 3.4)

Re-evaluating the Role of Memorization. Motivated by these issues, we introduce principled fixes. We use high data sample rates to reduce bias, prose a memorization definition based on fractional difference to reduce false positive, provide a mechanism for statistically grounding the thresholds, and employ test datasets specifically designed to avoid leakage. *Our experiments show that removing memorized at worst does not impact accuracy, and at best, increases test accuracy.* These results challenge the notion that memorization is necessary for generalization, and in fact show that their presence harms generalization. Given its well documented downsides, we argue that memorization should not be viewed as a desirable feature in deep learning models, but rather as a behavior to be mitigated.

2 Understanding Feldman and Zhang [12]

Having gone over how different factors influence memorization, we describe in detail the original work of Feldman and Zhang [12]. Our primary goal is to evaluate their methodology, recommend experimental fixes, and consequently, reassess their findings. To that end, we describe how they 1) define memorization, 2) approximate memorization scores, and 3) quantify marginal utility.

2.1 Defining Memorization

Feldman and Zhang [12] define a memorized point as one having high self-influence (i.e., a point that is predicted correctly only when present in the training data).

Specifically, consider a training set $S = ((x_1, y_1)...(x_n, y_n))$ and a point x_i in the training set S. The memorization score is the difference in prediction accuracy between when the point x_i is present in the training data ($h \leftarrow A(S)$) and when x_i is absent ($h \leftarrow A(S^{\setminus i})$). Here, ($h \leftarrow A(S)$) means that models h were trained on dataset S using algorithm A. The *absolute* difference is calculated using:

$$\mathbf{Pr}_{h \leftarrow A(S)}[h(x_i) = y_i] - \mathbf{Pr}_{h \leftarrow A(S^{\setminus i})}[h(x_i) = y_i] \tag{1}$$

The definition captures the intuition that a point x_i has been memorized if its prediction changes significantly when it is removed from the dataset. We include Table 3 for reference on the symbols used throughout the paper.

For example, consider training 1000 instances each of the models $h \leftarrow A(S)$ and $h \leftarrow A(S^{\setminus i})$. If the correct classification rate for x_i when it $h \leftarrow A(S)$ is around 90% (i.e., 900 out of the 1000 instances classified the point correctly). However, it falls significantly when $h \leftarrow A(S^{\setminus i})$ to 25%

(i.e., 250 out of the 1000 instances classified the point correctly). Due to the significant drop in self accuracy, this point has a high self-influence, and therefore, a high memorization score, specifically of $90\% - 25\% = 65\%$. This means that x_i is far more likely be classified correctly when it is present in the training data. In contrast, if there is no significant change in the classification rate, then it has a low memorization score. In this case, x_i will likely be classified correctly, whether or not it is present in the train set. As a result, the memorization score of a given point will be inversely proportional to its sub-population size: the larger the sub-population the smaller the memorization score. In the case of our hypothetical cat dataset, the pink cat (singleton) will have the highest memorization score, followed by the black cats (small-subpopulation) and the white cats (large sub-population).

Having defined memorization, the next step is to develop a methodology to identify memorized points from a dataset. A point is considered memorized based on its memorization score, calculated using Equation 1. The most *precise* way to compute this score is via the classic leave-one-out experiment. Here, we remove a single point from the training dataset, retrain the model on the remaining data, and test to see if the removed point is correctly classified. We have to run this experiment on all the points in the dataset to get the memorization score for each. Additionally, we have to repeat this model training process, for each point, multiple times to account for randomness introduced during training (e.g., the varying initialization, GPU randomness, etc.). Specifically, this would require training hundreds models for every point in the data. Considering data sets contain tens of thousands of points, this would require training millions of models. Therefore, running this experiment over a large dataset and model will require a large amount of resources and is therefore, computationally intractable.

2.2 Approximation Algorithm and Marginal Utility:

To overcome this limitation, Feldman and Zhang [12] propose a method to *approximate* the memorization scores. To that end, the authors employ the following steps. **Step 1. Sampling:** Instead of removing one point at a time, the authors randomly sample a fraction r of the points from the training set (originally of size n) and leave the remaining points out of training. The number of points used in training is then $m = r \cdot n$, $0 \leq r \leq 1$. In [12] the authors use $r = 0.7$ for their experiments. **Step 2. Training:** Each sample of points is then used to train an individual model. As a result, a random point x_i will be present in approximately $k \cdot r$ of the total trained models and will be absent from $k \cdot (1 - r)$ of them. **Step 3. Memorization Score Calculation:** The memorization scores are aggregated the results over both sets of models using Equation 1. **Step 4. Thresholding:** All the points that have a higher memorization score than some predetermined threshold are said to be memorized. **Step 5. Marginal Utility:** Having identified the memorized points, the authors now calculate their marginal utility (i.e., their impact on test accuracy).

3 Feldman and Zhang [12] Limitations

Having outlined how the original authors defined and approximated memorization, we now turn to examining the various sources of error at each of the steps. In each case, we demonstrate the source of error and how to fix it. Our goal is to (1) demonstrate how these errors undermine the central conclusion about the relationship between memorization and generalization and (2) provide solutions to fix these errors. Our experiments vary over different datasets (CIFAR-10/100, and Imagenet), models (MobileNet, VGG19, Resnet18, and Resnet50), and different training parameters. We use the same training setup as [11], with details in Appendix A. We train approximately 50 models for each dataset model pair. For succinctness, we present the RESNET-50 results in the main body of the work and leave the rest in the Appendix.

3.1 Sampling Bias leads to high approximation error of Memorization Scores

Step 1 in Feldman and Zhang [12]'s approximation algorithm (Section 2.2) is to take random samples of the data in order to train the models. Here, the choice of the sampling rate hyperparameter can significantly affect the accuracy of memorization scores. An incorrect choice can lead to sampling bias, resulting in a high approximation errors. Specifically, using **lower sampling rates**, i.e., training on a small subset of the data, can introduce significant approximation error due to overfitting. This is because there are not enough points for the model to learn the over-arching pattern. Instead, the model will memorize the few points it sees during training. As a result, both memorized

and non-memorized points will be marked as memorized. This does not mean that these points will be memorized by model when trained on the full dataset, but are an artifact of a low sample rate. In contrast, **higher sampling rates**, those approaching the leave-one-out (i.e., evaluating memorization by only removing one point at a time, the gold standard for memorization calculation [11]), will yield more reliable results. In the leave-one-out (LOO) setup, each model is trained on the entire dataset except for one held-out point, which is then evaluated. This ensures that any correct prediction for the held-out point is due to generalization from the remaining data, rather than direct memorization of its label [26] . Therefore, training at high sampling rates, i.e., training on large subsets of the data, will lead to more accurate results.

To empirically demonstrate the impact of sampling rate, we train ensembles of models using a range of sampling rates from 0.1 to 0.5. Since true LOO evaluation is computationally infeasible in our setting, we use a very high sampling rate (close to 0.95) as a practical baseline, which closely approximates LOO behavior. We then quantify the approximation error by comparing memorization scores with the baseline. Figure 1 and 6 show the results our experiments. We can see that across every model and dataset, as sampling rate increases, approximation error decreases. This analysis illustrates how lower sampling rates introduce significant bias, resulting in unreliable memorization measurements. Our findings provide empirical validation for the presence of sampling bias, and therefore, necessitate the the use of high rates of memorization calculation.

Unfortunately, the original work did not take this into account and employed a sampling rate of 0.7. On its own, using this sample rate does not necessarily disprove the findings of the original work. However, this rate does produce substantial approximation error in the memorization scores. As a result, points identified as highly memorized at this sampling rate may not in reality been memorized. As a result, the drop in accuracy after removing them might lead to the conclusion that memorization in needed for generalization. However, this would be flawed as the removed points contained non-memorized data as well. A precise experimental setup should only remove memorized points without removing the non-memorized data.

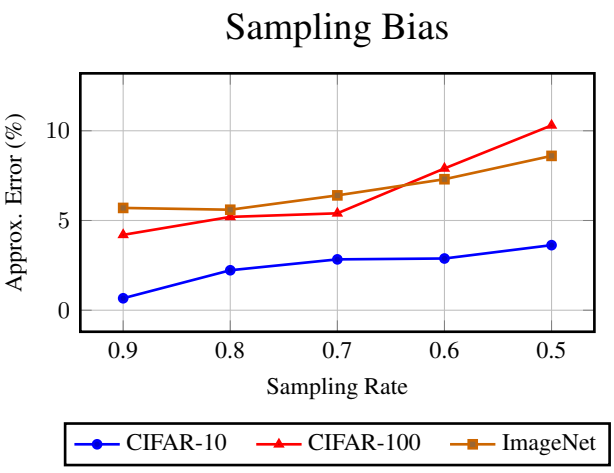


Figure 1: Lower the sampling rate leads to higher the approximation errors.

Proposed Solution: To ensure trustworthy conclusions, an accurate study of memorization vs generalization should use a sampling rate as close to leave-one-out as computational constraints permit.

3.2 The Original Definition has High False Positive Rates

Having trained the models from the sampled data, Feldman and Zhang [12], Step 3 of the approximation algorithm (Section 2.2) involves calculating the memorization scores using Equation 1. A good memorization definition should distinguish between points that are truly memorized (e.g., outliers) and those that generalize well (e.g., inliers). However, we hypothesize that the simple difference score defined in Equation 1 does not do so and leads to a high false positive rate when identifying memorized points.

Consider an outlier point x_o that is correctly classified by 25% of models when it is present in the training set, and by 0% of models when it is absent. According to the memorization score, its value is $25\% - 0\% = 25\%$. Now, consider an inlier point x_i that generalizes well across models: it is correctly classified 100% of the time when present, and 75% of the time when absent. Its memorization score is also $100\% - 75\% = 25\%$.

Despite their fundamentally different behaviors, both points receive the same memorization score. This contradicts the intuition that x_o, which is *only* learned when explicitly trained on, should be

considered more memorized than x_i, which is consistently learned even without training exposure. As a result, the score fails to meaningfully separate memorized points from generalizable ones. This results in a high false positive rate and undermines the reliability of the conclusions.

Instead of using the simple difference, as in Equation 1, we propose using *fractional* difference:

$$\frac{\mathbf{Pr}_{h\leftarrow A(S)}[h(x_i)=y_i]-\mathbf{Pr}_{h\leftarrow A(S^{\setminus i})}[h(x_i)=y_i]}{\mathbf{Pr}_{h\leftarrow A(S)}[h(x_i)=y_i]+\mathbf{Pr}_{h\leftarrow A(S^{\setminus i})}[h(x_i)=y_i]} \tag{2}$$

This updated scoring function[1] assigns the outlier x_o a higher memorization score of $\frac{25\%-0\%}{25\%+0\%} = 100\%$, while the inlier x_i receives a lower score of $\frac{100\%-75\%}{100\%+75\%} \approx 14.3\%$. This ensures that x_o and x_i are distinguishable under the revised definition, addressing the ambiguity in the original formulation.

Now, we empirically evaluate both the original and updated definitions in their ability to identify memorized points. To do so, we use randomly labeled points as a baseline for memorized points, similar to prior work [1, 5, 9, 10]. These points can only be correctly classified via memorization, as there is no underlying structure to generalize. The definition that makes fewer mistakes in distinguishing these memorized random-label points from natural examples is considered more effective. We quantify detection performance by randomly labeling 5% of the data. We then use the False Positive Rate at 95% True Positive Rate (FPR@95% TPR), a standard metric for evaluating detection accuracy. It is scaled between 0 and 1, with higher values corresponding to more mistakes (and therefore, poorer performance). We thoroughly evaluate our definition, we run experiments over varying conditions including different regularization parameters.

Figure 2 shows a subset of our results (with the full results in Figure 7a and Figure 8). Our results highlight two key findings. **First**, the original memorization definition results in consistently high FPR, often exceeding 25%, across all datasets. This indicates that the definition frequently fails to distinguish random-label points (which are truly memorized) from natural training data points (which may generalize). Such misclassifications casts doubt on the marginal utility findings of Feldman and Zhang [12], which employs on the original definition to identify memorized examples.

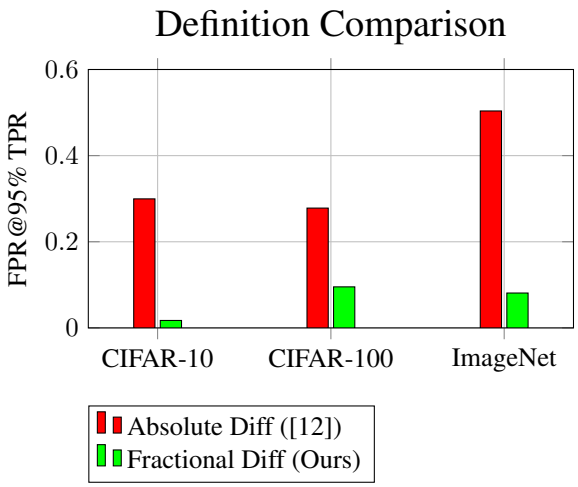


Figure 2: The original definition proposed by [12] has a high FPR (i.e., it makes many mistakes). However, we can reduce this significantly by using fractional difference as shown in Equation 2.

Second, our modified definition significantly reduces the FPR across all settings. This suggests that our method makes fewer mistakes when identifying truly memorized points. To further assess its robustness, we evaluate both definitions under varying conditions. These include evaluating over different regularization parameters (Figure 8) and comparing AUCs across the entire ROC curve (Figure 7a) and Precision-Recall curves (Figure 7b). We find that our modified definition consistently outperforms the original. This reinforces our claim that the revised scoring function better captures the true nature of memorization.

Proposed Solution: These findings underscore that the original definition is unreliable, due to its high FPR. As a consequence, it is imperative to use a low-FPR scoring function (specifically the one we propose) when estimating the marginal utility of memorized points. Without a reliable definition, non-memorized points may be mistakenly grouped with memorized ones, leading to flawed conclusions about the relationship between memorization and generalization.

[1]We considered two alternative denominators: either $\Pr_{h\leftarrow A(S)}[h(x_i)=y_i]$ or $\Pr_{h\leftarrow A(S^{\setminus i})}[h(x_i)=y_i]$. However, using the sum of the two terms is known to provide greater numerical stability [28], and was therefore selected as our final choice.

3.3 Unprincipled Memorization Threshold Leads to Inconsistent Results

Once the memorization scores have been calculated, Step 4 of the approximation algorithm (Section 2.2) is used to decide how high the memorization scores need to be to mark the points as memorized? While Feldman and Zhang [12] uses a threshold of 0.25, they do not provide any evidence for this value.

This is an issue, as their findings, and that of their earlier theoretical work [11], are based on the *premise* that Equation 1 provides a precise representation of memorized points (Section 2.1). We argue that the key limitation is not in their derived proofs, but in absence of a statistically robust threshold as the cornerstone of their work. For example, points with high memorization scores can behave in contradictory ways to ones with low memorization scores. Specifically, the authors show that removing memorized points with high scores (i.e., using a high memorization threshold) does not impact test accuracy. However, removing memorized points with low scores does impact test accuracy. Therefore, conclusions derived from one threshold can easily be disproved with points from a different threshold. A natural question arises: which one of the two thresholds really represents the memorized data? Do we base our conclusion on the high thresholds and claim that memorization does not impact accuracy? Or based it on the low thresholds and argue the opposite?

We answer this question in a statistically robust manner. Influenced from earlier works [1, 5, 9, 10], we propose using a *random-label calibration distribution* of memorization scores derived from randomly labeled training points. Specifically, let $\mathcal{S}_{\text{rand}}$ denote the random-label calibration distribution or the set of memorization scores computed for randomly labeled data points, and let $s(x)$ be the memorization score of a natural (i.e., non-random) data point x. We define a threshold τ based on the $(1-\alpha)$-quantile of the random-label calibration distribution:

$$\tau = \text{Quantile}_{1-\alpha}(\mathcal{S}_{\text{rand}}) \tag{3}$$

We then classify a data point x as memorized if $s(x) > \tau$. For example, setting $\alpha = 0.05$ corresponds to selecting the 95th percentile of $\mathcal{S}_{\text{rand}}$ as the threshold. This ensures that only those points with memorization scores significantly higher than what would be expected under random labeling are identified as memorized. This method provides a principled and interpretable statistical criterion for distinguishing memorization from generalization.

Figure 3 provides an illustration of our proposed method for identifying memorized points using a random-label calibration distribution. In this conceptual example, the blue curve represents the distribution of memorization scores for randomly labeled training points, while the orange curve shows scores for natural points. The red dashed line denotes the 95th percentile of the random-label calibration distribution, which serves as the threshold for identifying memorization. Any point with a score exceeding this threshold is classified as memorized with high statistical confidence. While the distributions in this figure are synthetic, the example demonstrates how our method cleanly separates memorized points from those that generalize, and highlights the importance of grounding memorization definitions in statistically meaningful baselines.

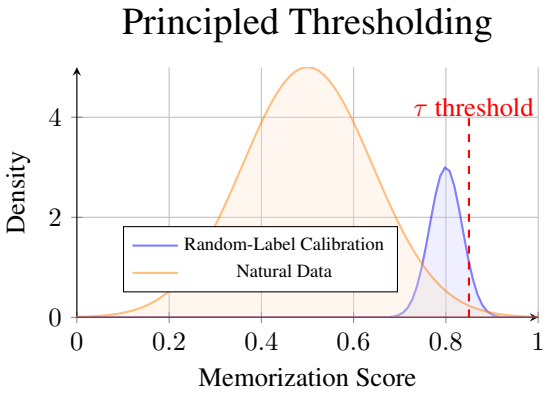


Figure 3: Illustration of our proposed null-based memorization threshold.

Proposed Solution: In the next section, when we evaluate the relationship between and generalization, we employ this statistical threshold to identify memorization. Additionally, we run further experiments using decreasing thresholds (like the original work) for completeness.

3.4 Data Leakage leads to overestimation of test accuracy

Once the points are identified, Feldman and Zhang [12] now calculate the marginal utility of the memorized points by quantifying test set accuracy in the final step (Section 2.2). However, here too, we find the presence of a common oversight from the ML literature, known as data leakage. This is when there is overlap between training and test data, such as the presence of duplicate or near-duplicate example and is a well-known threat to the validity of generalization estimates [14]. It has been documented in several popular datasets used in the literature, including CIFAR and ImageNet [17]. Data leakage artificially inflates test accuracy by allowing models to "peek" at the test set during training, creating the illusion of strong generalization. However, this is misleading, as true generalization is defined by performance on *unseen* data.

To assess whether data leakage plays a role in the memorization study by Feldman and Zhang [12], we conducted a preliminary investigation of the CIFAR-100 dataset using influence scores from [12]. We found numerous cases where training points with high memorization scores had near-duplicate counterparts in the test set, shown in Figure 4. In total, CIFAR-10 and 100 contain the presence of 8% [17] and 10% [2] duplicates.

This observation raises a crucial point: removing memorized points may reduce test accuracy not because memorization is essential for generalization, but because those points directly overlap with duplicates in the test set. In other words, the accuracy drop may reflect the removal of train-test duplicates, rather than a loss of generalization ability. This issue is further exacerbated by the fact that such duplicates constitute a substantial fraction of the training data, meaning that their removal disproportionately eliminates samples that the model is evaluated on verbatim, thereby artificially depressing test accuracy.



Figure 4: Data leakage: training points with high memorization scores that have near-duplicate counterparts in the test set.

Proposed Solution: While this experiment does not fully resolve the relationship between memorization and generalization (which we explore further in the next section), it highlights the importance of eliminating data leakage when interpreting results.

4 Re-Evaluating Marginal Utility

Having identified the different sources of error, we now implement the necessary fixes, identify the correct memorized points, and observe their impact on model accuracy.

4.1 Setup:

In order to perform the fairest evaluation, we re-run the experiments from the original paper and employ a similar experimental setup to the original paper (Details in Appendix A). To avoid sampling bias, we use a very high sampling rate (0.95) in order to minimize the approximation error. We use fractional difference (Equation 2) to avoid false positives when identifying memorization scores. We use our principled thresholding method to identify the memorized points for each dataset CIFAR-10, CIFAR-100, and ImageNet. Finally, to avoid data leakage, we use the test datasets that were purposefully designed to avoid train-test duplicates from previous works [2, 18]. We provide a outline of their methodology in Appendix A.2.

To evaluate whether memorization is necessary for generalization, we train models under each of the two conditions, identical to the original Feldman and Zhang [12] work: 1) Memorized-point removal: all identified memorized points were removed from the training set. 2) Random-point removal: an equal number of training points were removed at random, serving as a control. This ensures that any observed differences in performance are attributable to the specific removal of memorized points rather than simply training with fewer data. We report the difference in test accuracy

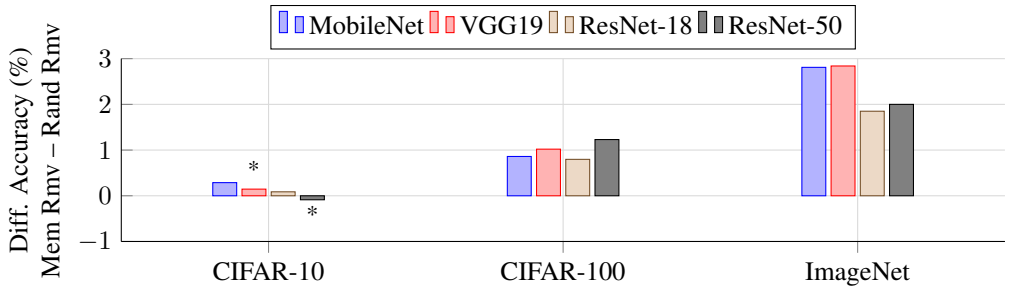


Figure 5: Change in accuracy when removing memorized points vs an equal number of random ones. * indicates that the change is not statistically significant (via a two-sided t-test), meaning the distributions are identical. Our results show the impact of removing memorized points: performance is unchanged at worst and significantly improved at best, especially on CIFAR-100 and ImageNet.

between the two conditions (test accuracy when memorized points removed vs. test accuracy when random points removed). To assess statistical reliability, we conducted paired t-tests across multiple training runs for each dataset, reporting the two-sided p-values. We conduct our experiments over four different models to validate our results.

4.2 Results:

Our results can be seen in Figure 5. As a reminder, we report the difference in test accuracy between removing memorized points and removing random points, following the original work. Positive values indicate that removing memorized points improves test accuracy (and vice-versa). We provide exact scores and p values for the associated t-test in the Appendix (Table 4). There three major takeaways from out results. **In most cases, removing memorized points improves accuracy:** In most cases (9 out of 12), removing memorized points yields a statistically significant increase in accuracy. This effect is observed across all models trained on ImageNet and CIFAR-100, and for MobileNet on CIFAR-10. For example, CIFAR-100 and ImageNet, we can observe an average increase of 0.5 % and 1.2%. In each of these cases, the improvement is statistically significant, with t-tests producing p-values of less than 0.05 (Table 4). This means that removing memorized points improves accuracy in most settings. **In the remaining cases, removing memorized points has no effect:** In 3 of the 12 cases (entirely in CIFAR-10), we observe that removing memorizing points leads to a small statically-insignificant change in accuracy, indicated by * in Figure 5. The t-tests produce p-values are greater than 0.05 (Table 4). This means that though the differences might either positive or negative, they are not statistically significant and can be considered equivalent to zero.

5 Are Memorized Points Just Mislabeled?

A natural concern is whether the points identified as memorized by our method are simply mislabeled or noisy examples. If so, the observed improvement in accuracy after removing these points would be trivially explained by the removal of label noise, rather than by eliminating genuine memorization. We conduct several analyses to investigate this possibility.

CIFAR-10 Mislabel Overlap. We compare our identified memorized points against approximately 4,500 potentially mislabeled examples in CIFAR-10 identified by prior work [27]. Among the approximately 750 memorized points removed by our method, only 177 (24%) overlap with this set. This suggests that the majority of removed points are not simply mislabeled examples.

ImageNet CLIP-based Audit. To assess label quality on ImageNet, we pass all images identified as memorized through CLIP and check whether the dataset label appears among CLIP's top-N predictions. Among the approximately 21,000 images we identify as memorized, around 3,200 (16%) do not appear in CLIP's top-5 predictions, and only 420 (3%) do not appear in the top-10. This indicates that mislabeled examples account for a small fraction of the memorized set.

Control Experiment: Excluding Mislabeled Samples. To further isolate the effect of mislabeled examples, we retrain ResNet-50 models on ImageNet after excluding the approximately 3,100 likely mislabeled examples identified by the CLIP audit. After this exclusion, removing memorized points

still yields a positive marginal utility of +0.9%, compared to +1.98% when mislabeled examples are included. This confirms that the improvement in accuracy is not fully explained by the removal of noisy labels.

Training Dynamics Analysis. Finally, we examine whether memorized points exhibit distinct training dynamics compared to mislabeled examples. We compare four groups of ImageNet training examples: (1)*real*$_{inliers}$: examples consistently classified correctly regardless of inclusion in training; (2) *points*$_{100\to75}$: examples with $P_{in} = 1.0$, $P_{out} = 0.75$, marked as memorized by FZ's absolute-difference metric; (3) *points*$_{25\to0}$: examples with $P_{in} = 0.25$, $P_{out} = 0$, receiving the same FZ score but a much higher fractional score; and (4)*real*$_{noise}$: examples with randomly flipped labels. For each group, we measure learning speed [1], forgetting score [24], and logit score [23]. Results are shown in Table 1.

Table 1: Training dynamics of different point sets on ImageNet.

Point Set	Learning Score	Forgetting Score	Logit Score
real_inliers	68	0.02	50
points$_{100\to75}$	67	0.02	49
real$_{noise}$	64	0.04	46
points$_{25\to0}$	25	0.12	42

Two observations emerge. First, *points*$_{100\to75}$ behave nearly identically to *real*$_{inliers}$, suggesting they do not correspond to memorized subpopulations as FZ assert. Second, *points*$_{25\to0}$ exhibit distinct dynamics—learned slowest, forgotten most, and with lowest logit values—yet they do not behave like randomly mislabeled*real*$_{noise}$ points. This provides further evidence that the points identified by our fractional-difference metric are genuinely memorized, not merely mislabeled.

6 Ablation Study

Having shown that removing memorized points improves model accuracy, we now perform an ablation study to see which one of the four factors plays the largest role.

6.1 Setup

We employ the experimental setup described in Section 4. Specifically, we identify memorized points on CIFAR-10 and 100 using ResNet-50 models trained with the same criteria as in Feldman et al. We use a sampling rate of 0.7 and mark points with a memorization score above 0.25 (under the original definition) as memorized. We then remove these memorized points from the training set, retrain the model, and evaluate test accuracy on the original test set (i.e., without de-duplication). This configuration serves as our baseline. Next, we introduce one improvement at a time to each of the four factors considered in our work (sampling bias, fractional-difference definition, principled thresholding, and data leakage) and, for each modified configuration, retrain the model and measure the resulting change in utility (over 50 models) and improvement over the baseline. Finally, we run the two sided t-test to see which factors introduce a statically significant improvement in utility over the baseline.

6.2 Results

Our results are shown in Table 2. We can make a few observations: First, combining all four components usually provides the strongest results. The "All Components Changed" row shows that the combined system almost always outperforms baseline across all datasets. This implies the corrections are additive rather than redundant, meaning each modification addresses a different failure mode in the original Feldman setup. Second, the four factors do not contribute equally across each data. For example, data de-duplication and principled thresholding play a much stronger role than fractional-difference. Third, the impact of all components is grows as dataset complexity increases. This is reflected in the Δ values for the "All Components Changed" row, which rise from 1.59 on CIFAR-10 to 2.22 on CIFAR-100 and 7 on ImageNet. Thus, the gap between the revised setup and the original Feldman and Zhang setup becomes more pronounced on larger and more

Table 2: The results of the ablation study. Marginal Utility (MU) is the difference in accuracy between models trained with memorized points removed vs random removal. Δ is the difference in utilities between the new setup and the Feldman and Zhang one.

	CIFAR-10		CIFAR-100		ImageNet	
Setting	MU	Δ	MU	Δ	MU	Δ
Baseline (Feldman and Zhang setup)	-1.71	-	-1.06	-	-4.98	-
+ Fractional Difference	-1.31	0.4	-1.17	-0.11	-5.04	-0.02
+ 0.95 Sampling Rate	-1.15	0.56	-0.44	0.6	-1.68	3
+ Principled Thresholding	0.12	1.84	-0.04	1.02	-0.59	4.39
+ DeDuplicated Test	0.006	1.72	0.02	1.09	1.09	6.08
All Components Changed	**-0.11**	**1.59**	**1.15**	**2.22**	**1.97**	**6.96**

complex datasets. Finally, the ablations reveal an approximate trend from weakest to strongest effect across nearly all experiments: fractional difference contributes the least, increasing the sampling rate to 0.95 provides a moderate gain, principled thresholding gives a stronger correction, and test-set de-duplication contributes the largest improvement overall.

Overall, our ablation shows that the failure mode of the original Feldman and Zhang [12] method is not due to any single component, but rather to the cumulative effect of several small sources of error. Individually, each fix only partially reduces the negative utility, but together they correct the underlying instability and reverse the sign of the effect. Once these sources of error are corrected, we find that removing memorized points improves generalization.

7 Discussion and Conclusion

Correcting Misconceptions from Prior Work: Our results show that removing memorized points never harms performance (CIFAR-10) and can also significantly improve accuracy on more complex datasets (CIFAR-100, ImageNet). This directly contrasts with the original work [12], which reported that removing memorized points reduced accuracy. The discrepancy arises because their analysis suffered from four methodological flaws (Section 3): sampling bias, a high false positive rate, unprincipled thresholding, and data leakage. Correcting for these issues reveals that memorized points are **not** necessary for generalization; on the contrary, their removal can improve test performance.

Commonly Overlooked Errors: The four errors we identify in our work sampling bias, flawed definitions, unprincipled thresholding, and data leakage, are not unique to Feldman and Zhang [12], but can be considered common oversights in machine learning. Each issue stems from methodological shortcuts that, while convenient, can seriously distort the final conclusions. As we see here, a small methodological lapses risk compounding into broad conceptual errors in our understanding.

Memorization and Privacy: Memorization has a direct implication for security and privacy research. This is because one hand, Feldman [11] claim that memorization was needed for generalization while other works demonstrated that memorization leads to lead to leakage of private data, model stealing attacks, increased model bias, poor robustness to distribution shifts, and susceptibility to adversarial samples. In other words, generalization and security/privacy can not be simultaneously achieved. While this might have dissuaded some researchers, our work shows that this tension does not exist. This is because memorization is not necessary for generalization. Future researchers are encouraged to explore methods to build models that both generalize and are private.

Limitations: Our approach requires training approximately 1,000 models per dataset to accurately approximate leave-one-out scores, which is computationally expensive and impractical for larger datasets or models. Alternatively, once could explore other methods to approximate memorization scores. These include influence functions [15] or other data attribution methods [**? ?**]. However, these too are just estimates, and do not capture the precise scores for such a study, which estimate the effect of removing a training point without retraining. Additionally, our experiments focus on image classification. Extending this analysis to other modalities such as large language models, where memorization poses significant privacy risks, is an important direction for future work.

References

[1] Devansh Arpit, Stanisław Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks. In *International Conference on Machine Learning*, 2017.

[2] Björn Barz and Joachim Denzler. Do we train on test data? purging cifar of near-duplicates. *Journal of Imaging*, 6(6):41, 2020.

[3] Reza Bayat, Mohammad Pezeshki, Elvis Dohmatob, David Lopez-Paz, and Pascal Vincent. The pitfalls of memorization: When memorization hurts generalization. *arXiv preprint arXiv:2412.07684*, 2024.

[4] Emily Black and Matt Fredrikson. Leave-one-out unfairness. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 285–295, 2021.

[5] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 267–284, 2019.

[6] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650, 2021.

[7] Nicholas Carlini, Matthew Jagielski, Chiyuan Zhang, Nicolas Papernot, Andreas Terzis, and Florian Tramer. The privacy onion effect: Memorization is relative. *Advances in Neural Information Processing Systems*, 35:13263–13276, 2022.

[8] Nicolas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramer, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 5253–5270, 2023.

[9] Satrajit Chatterjee. Learning and memorization. In *International conference on machine learning*, pages 755–763. PMLR, 2018.

[10] Satrajit Chatterjee. Coherent gradients: An approach to understanding generalization in gradient descent-based optimization. *arXiv preprint arXiv:2002.10657*, 2020.

[11] Vitaly Feldman. Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 954–959, 2020.

[12] Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. *Advances in Neural Information Processing Systems*, 33: 2881–2891, 2020.

[13] Bargav Jayaraman, Lingxiao Wang, Katherine Knipmeyer, Quanquan Gu, and David Evans. Revisiting membership inference under realistic assumptions. *arXiv preprint arXiv:2005.10881*, 2020.

[14] Shachar Kaufman, Saharon Rosset, Claudia Perlich, and Ori Stitelman. Leakage in data mining: Formulation, detection, and avoidance. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6(4):1–21, 2012.

[15] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pages 1885–1894. PMLR, 2017.

[16] Yunhui Long, Lei Wang, Diyue Bu, Vincent Bindschaedler, Xiaofeng Wang, Haixu Tang, Carl A Gunter, and Kai Chen. A pragmatic approach to membership inferences on machine learning models. In *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 521–534. IEEE, 2020.

[17] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do cifar-10 classifiers generalize to cifar-10? *arXiv preprint arXiv:1806.00451*, 2018.

[18] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International conference on machine learning*, pages 5389–5400. PMLR, 2019.

[19] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. *arXiv preprint arXiv:1806.01246*, 2018.

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

[21] Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Diffusion art or digital forgery? investigating data replication in diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6048–6058, 2023.

[22] Cory Stephenson, Suchismita Padhy, Abhinav Ganesh, Yue Hui, Hanlin Tang, and SueYeon Chung. On the geometry of generalization and memorization in deep neural networks. *arXiv preprint arXiv:2105.14602*, 2021.

[23] Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A Smith, and Yejin Choi. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Empirical Methods in Natural Language Processing*, 2020.

[24] Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. An empirical study of example forgetting during deep neural network learning. In *International Conference on Learning Representations*, 2019.

[25] Dmitrii Usynin, Moritz Knolle, and Georgios Kaissis. Memorisation in machine learning: A survey of results. *Transactions on Machine Learning Research*, 2024.

[26] V.N. Vapnik. An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, 10(5):988–999, 1999. doi: 10.1109/72.788640.

[27] Jiaheng Wei, Zhaowei Zhu, Hao Cheng, Tongliang Liu, Gang Niu, and Yang Liu. Learning with noisy labels revisited: A study using real-world human annotations. In *International Conference on Learning Representations*, 2022.

[28] Guo-Cheng Wu and Dumitru Baleanu. Stability analysis of impulsive fractional difference equations. *Fractional Calculus and Applied Analysis*, 21(2):354–375, 2018.

[29] Jiayuan Ye, Aadyaa Maddi, Sasi Kumar Murakonda, Vincent Bindschaedler, and Reza Shokri. Enhanced membership inference attacks against machine learning models. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 3093–3106, 2022.

[30] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *International Conference on Learning Representations (ICLR)*, 2017.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

1. **Claims**

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

   Answer: [Yes]

   Justification: Yes.

   Guidelines:

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

2. **Limitations**

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

   Answer: [No]

   Justification: Not discussed.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might

not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.

- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

   Justification: Not applicable.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Yes.

   Guidelines:

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

(a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [N/A]

   Justification: Not applicable.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Reported.

   Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [Yes]

Justification: Reported.

Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [N/A]

   Justification: Not applicable.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: Yes.

   Guidelines:

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

10. **Broader impacts**

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

Answer: [N/A]

Justification: Not applicable.

Guidelines:

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

11. **Safeguards**

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

Answer: [N/A]

Justification: Not applicable.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: Authors cited.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.

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

13. **New assets**

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

    Answer: [N/A]

    Justification: Not applicable.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: Not applicable.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: Not applicable.

Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: Not applicable.

    Guidelines:

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

A Appendix

A.1 Training Setup

Our training setup exists of using 100 epochs (CIFAR-10 and 100) and 75 epochs (ImageNet), using a batch size of 512, with a triangular learning rate of 0.4. We train an average of 275 models for sampling bias experiments (Figure 1) and definition comparison experiments (Figure 2). We find this is sufficient to produce consistent results. When comparing definitions, we select points that are absent from at least approximately 10 models. This is done to ensure that a point that is absent from just a single model does not bias the resulting score calculations. However, we train substantially more models when the sampling rate is 0.95. Approximately 1,000 models for CIFAR-10/100 and ImageNet. We found that training these numbers of models was sufficient since memorization scores tend to stabilize much earlier (Figure(9). To calculate marginal utility, we train 100 models for Cifar-10/100 and 50 for ImageNet. This too is sufficient since the model accuracy has stabilized at this point (Figure 11).

A.2 Data Leakage Removal

We did not perform our own near-duplicate detection; instead, we relied on the deduplicated test sets released by [2] and [17], both of whom identify duplicates using cosine similarity in deep feature space rather than pixel-level distance. [2] report up to 3% duplicates in CIFAR-10 and up to 10% in CIFAR-100, while [17] later found that CIFAR-10 contains roughly 8% duplicates, substantially more than previously reported, and manually curated fully deduplicated test sets for CIFAR-10 and ImageNet to eliminate this issue. Because no curated CIFAR-100 set was released by [17], we adopt the deduplicated version from [2] and do not recompute similarity thresholds, per-class removal counts, or pixel-space statistics, since doing so would introduce a different notion of duplication and risk class-specific bias.

Table 3: Symbols used and their meanings.

Symbol	***Meaning***
x_i	training data point
y_i	training point label
x_i'	test data point
y_i'	test point label
S	training set
A	training algorithm
n	size of the training set
m	number of points removed from the training set
h	trained model
t	trial

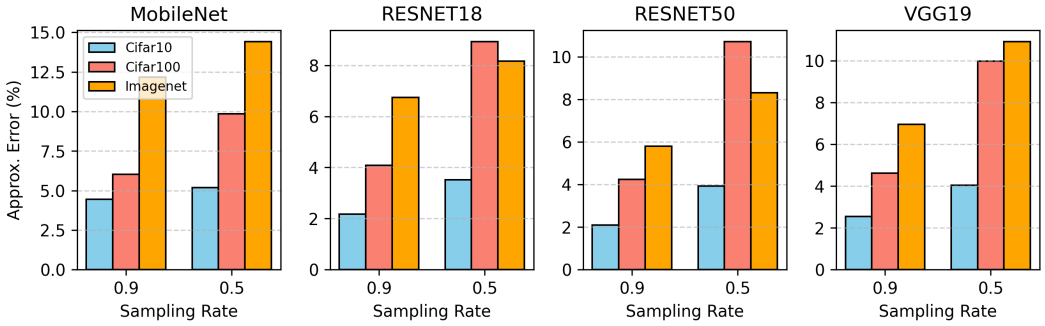
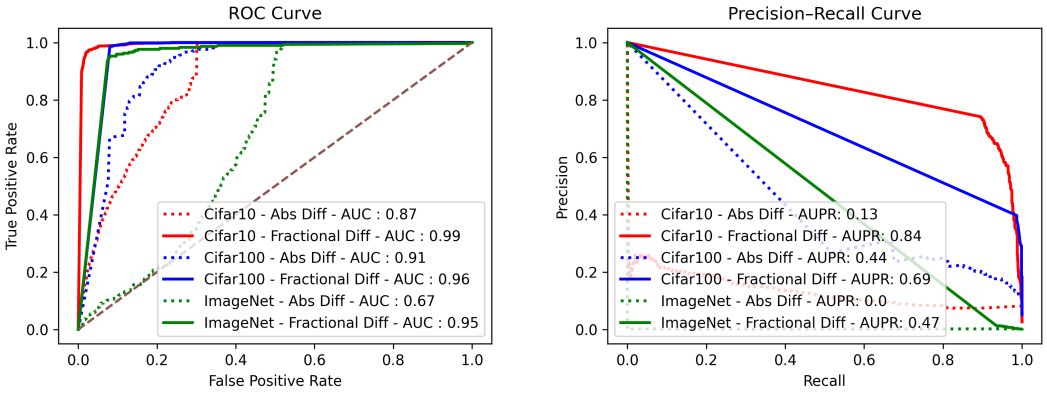


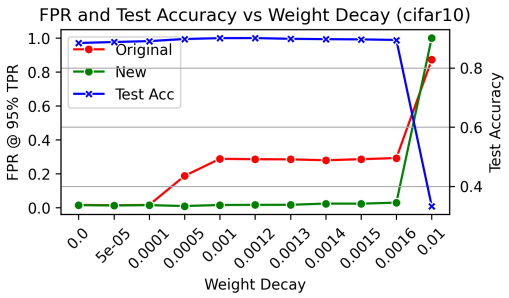
Figure 6: Sampling bias: Higher sampling rates result in more accurate scores. While lower rates lead to higher approximation error.



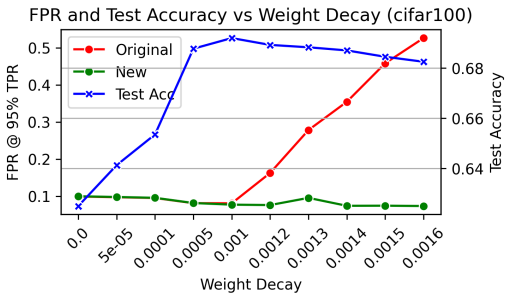
(a) ROC curves (AUROC).

(b) Precision–recall curves (AUPR).

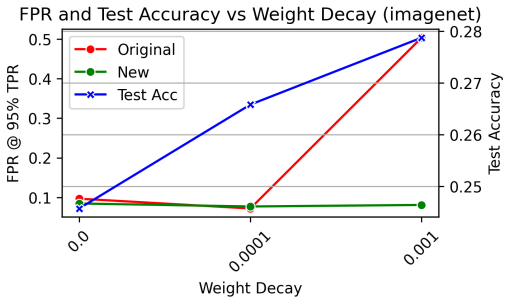
Figure 7: Memorization definitions evaluated via ROC and precision–recall curves on CIFAR-10, CIFAR-100, and ImageNet. Our fractional-difference definition consistently outperforms the original score, achieving higher AUROC and AUPR across all datasets.



(a) CIFAR-10



(b) CIFAR-100



(c) ImageNet

Figure 8: Visual comparison of results across datasets. The final quadrant is left intentionally blank.

Table 4: Comparison of model accuracy (in %) after removing memorized samples versus randomly selected samples, evaluated across datasets (CIFAR-10, CIFAR-100, ImageNet) and architectures (MobileNet, VGG19, ResNet18, ResNet50).

Dataset	Model	Mem Remove Acc (%)	Rand Remove Acc (%)	Difference ($\pm$ std)	p-value	Significance
CIFAR-10						
	MobileNet	81.34	80.98	0.36 ± 1.29	2.18×10^{-2}	Significant
	VGG19	89.91	89.61	0.29 ± 2.40	1.92×10^{-1}	Not Significant
	ResNet18	91.19	91.12	0.07 ± 0.29	3.29×10^{-2}	Significant
	ResNet50	91.87	91.95	-0.07 ± 0.36	9.28×10^{-1}	Not Significant
CIFAR-100						
	MobileNet	41.34	41.23	0.22 ± 3.06	3.1×10^{-1}	Not Significant
	VGG19	60.67	59.67	0.99 ± 0.44	2.87×10^{-31}	Significant
	ResNet18	65.18	64.33	0.84 ± 0.40	4.18×10^{-25}	Significant
	ResNet50	67.86	66.78	1.07 ± 0.97	1.97×10^{-12}	Significant
ImageNet						
	MobileNet	29.68	26.87	2.81 ± 1.199	5.20×10^{-34}	Significant
	VGG19	43.31	35.17	8.14 ± 10.602	1.13×10^{-7}	Significant
	ResNet18	49.99	48.14	1.85 ± 0.861	4.75×10^{-30}	Significant
	ResNet50	53.38	50.63	2.75 ± 4.303	2.58×10^{-5}	Significant

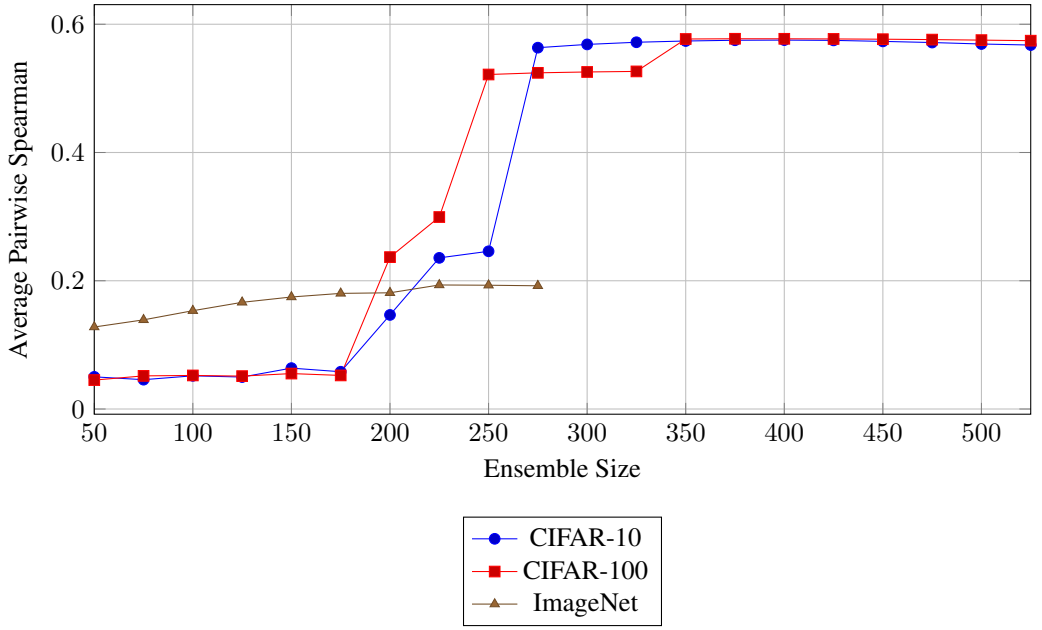


Figure 9: Average pairwise Spearman correlation between memorization scores as a function of ensemble size. The figure indicates that memorization scores stabilizes around 275, 350, and 225 models for CIFAR-10, 100, and Imagenet.

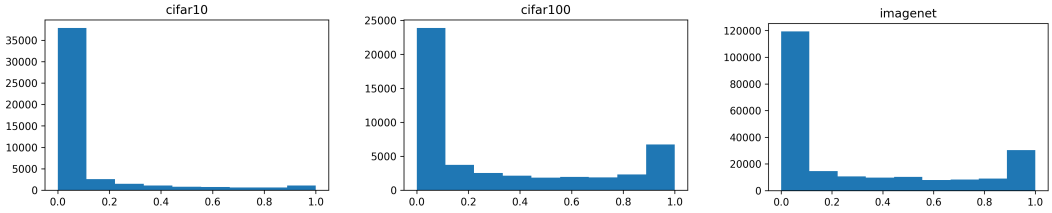
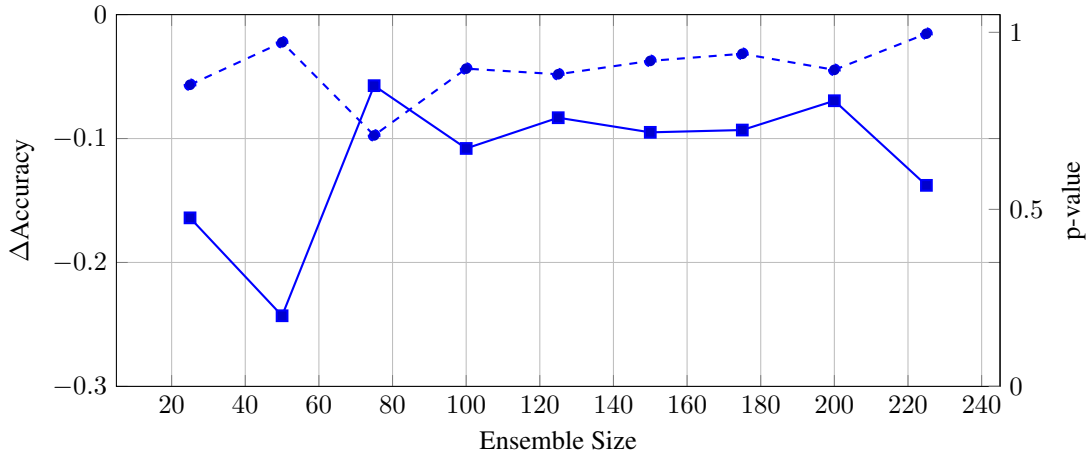
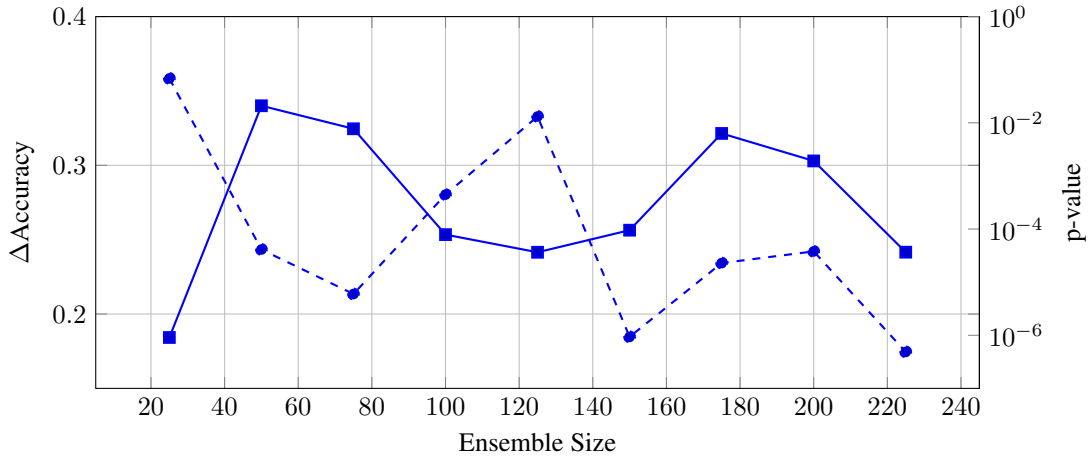


Figure 10: Distribution of memorization scores after removing the identified memorized points. We can see that the data distributions still show the long tailed structure, which is a necessary assumption for Feldmen et al's works.



(a) CIFAR-10: The plot shows how the difference in accuracy and the corresponding p-values vary with ensemble size. We can observe that p-value remains constantantly high across model ensembles indicating that removing memorized points does not impact model accuracy. The solid lines shows change in test accuracy and dotted line shows change in train accuracy.



(b) CIFAR-100: The plot shows how the difference in accuracy and the corresponding p-values vary with ensemble size. We can observe that p-value remains constantantly low across model ensables indicating that removing memorized points does improves model accuracy.

Figure 11: Ensemble size vs Utility