Evaluating Inexact Unlearning Requires Revisiting Forgetting

Shashwat Goel* Ameya Prabhu* Ponnurangam Kumaraguru
shashwatnow@gmail.com, ameya.prabhu@mailfence.com, pk.guru@iiit.ac.in
Code: https://github.com/shash42/Evaluating-Inexact-Unlearning

Abstract

Existing works in inexact machine unlearning focus on achieving indistinguishability from models retrained after removing the deletion set. We argue that indistinguishability is unnecessary, infeasible to measure, and its practical relaxations can be insufficient. We redefine the goal of unlearning as forgetting all information specific to the deletion set while maintaining high utility and resource efficiency.

Motivated by the practical application of removing mislabelled and biased data from models, we introduce a novel test to measure the degree of forgetting called Interclass Confusion (IC). It allows us to analyze two aspects of forgetting: (i) memorization and (ii) property generalization. Despite being a black-box test, IC can investigate whether information from the deletion set was erased until the early layers of the network. We empirically show that two simple unlearning methods, exact-unlearning and catastrophic-forgetting the final $k$ layers of a network, scale well to large deletion sets unlike prior unlearning methods. $k$ controls the forgetting-efficiency tradeoff at similar utility. Overall, we believe our formulation of unlearning and the IC test will guide the design of better unlearning algorithms.

1. Introduction

There have been growing concerns about the ethics of data use for designing machine learning systems, with primary considerations being privacy [44] and harmful data [20]. Privacy protections come forth with legislation like GDPR [19], CCPA [2], and PIPEDA [4]. These policies require organizations to allow users to delete their data from all systems, including learnt models [3, 21, 26] which can leak information about user data [15, 57]. Similarly, there has been a growing need to tackle problems in datasets ranging from random noisy labels [23, 49, 50] to systematic biases $^1$ [52], besides adversarial subsets such as poisoned data [9, 18, 69].

Removing the influence of the sensitive data points from trained models is a reliable approach to tackle both problems, user privacy requests and harmful data. The domain of machine unlearning [11, 14, 26] targets this data deletion problem. Formally, we are given a model $M$ trained on a dataset $D$ by a training procedure $T$, and any deletion set $D_f \subset D$. $^2$ The goal is to propose an unlearning procedure that generates a model $M_u$ by deleting any information present in $D_f$ but not in retain set $D \setminus D_f$. Our work is focused on inexact unlearning, where information removal may be imperfect or unproven.

As illustrated in Figure 1, most prior work in inexact unlearning formalizes the definition of unlearning as producing a model with a parameter [25, 31, 41, 65, 68] or output [10, 27–29, 48, 51] distribution indistinguishable from one produced by retraining from scratch using $T$ on $D \setminus D_f$ ($M_u$ and $M_r$ in Figure 1 respectively). We argue model indistinguishability is not necessary, infeasible to measure, and its existing relaxations are insufficient. We redefine unlearning from achieving model indistinguishability to having

\[ D \setminus D_f \]

\[ D \]

\[ \text{Training Procedure} \]

\[ \text{Retrained Model } M_r \]

\[ \text{Original Model } M \]

\[ \text{Information in } D_f \text{ but not in } D \setminus D_f \]

\[ \text{Prior Definition } M_r \text{ “Indistinguishability from Retrained Model } M_r^\prime \]

\[ \text{Our Unlearning Definition } M_u \text{ “No additional Information from } D_f^\prime \]

\[ \text{Figure 1. Difference between the goals of indistinguishability and our definition of forgetting, with each colour representing information contributed by different sets of data. Prior work only defines } M_u^\prime \text{ to have unlearnt. In contrast, we define unlearning as removing all information gained from } D_f \text{ (in black) which is achieved by both models } M_u \text{ and } M_u^\prime. \]

\[ ^2 D_f \text{ may not be i.i.d sampled from } D. \]

\[ ^1 \text{to the extent that bias is a dataset problem [37].} \]

\[ ^* \text{Equal Contribution} \]
three goals: forgetting, utility and resource efficiency, with a focus on forgetting. Figure 1 illustrates the forgetting criteria, which considers any unlearning procedure that has deleted information present in $D_f$ but not in $D \setminus D_f$ as having forgotten perfectly. Inexact-unlearning methods should be on the pareto frontier of the trade-off among the above three desiderata, preferably scaling to large deletion sets.

Any evaluation method for unlearning consists of a test defining which samples are removed and a metric to measure forgetting. We propose a black-box test called Interclass Confusion (IC) which induces confusion between two classes, requiring the unlearning procedure to erase this confusion. By doing this, the IC test enhances the gap between models which have unlearnt the deletion set to varying extents. We also propose the Forgetting Score (Fgt) as a metric that measures memorization \cite{7} when computed on the outputs of any model for $D_f$ and property generalization \cite{46} on unseen samples similar to $D_f$. The IC test and Fgt metric together significantly outperform existing evaluation methods in their ability to reliably demonstrate whether the information was removed from even the early layers of the network. In deep networks, we find simply performing unlearning on the final linear layer \cite{10, 39, 68} fails to remove most information.

Our forgetting-oriented framework allows us to analyze diverse phenomena. First, we empirically find that catastrophically forgetting \cite{24} from the final $k$ layers almost entirely removes their information while having a significantly better trade-off on time than retraining those layers. Second, we find that using some regularization methods can make the original model more amenable to unlearning. Thus, any comparison of unlearning efficacy must be attentive to the optimization procedure used to obtain $M$.

Overall, we alleviate the shortcomings of the model indistinguishability definition, weak evaluations methods and tiny deletion sets in prior work. Our main contributions are:

1. We propose a new definition of inexact-unlearning, which focuses on forgetting information contributed by the deletion set $D_f$.
2. We study the setting of large deletion set sizes. We show the resource efficiency of data-influence isolation, the prevalent strategy for exact-unlearning, scales poorly as deletion sets grow.
3. We propose the IC test which amplifies the difference in the information present in the original dataset $D$ and retain $D \setminus D_f$, and the Fgt metric which provides a targeted measurement of forgetting.
4. We investigate how early in a deep network is sample-specific information retained, the phenomenon of catastrophic-forgetting and possibilities for optimizing the original model for better unlearning.

2. Problem Formulation

This section presents our machine-unlearning formulation and discusses the shortcomings of the two most popular unlearning paradigms in existing literature: (i) Model indistinguishability (ii) Data-influence isolation. We define unlearning as having three critical goals: Forgetting, Utility and Resource Efficiency.

**Forgetting:** Given any subset of the data $D_f \subset D$, forgetting is defined as removing the information the subset $D_f$ contributes to the model which cannot be obtained from $D \setminus D_f$.

**Utility:** A naive unlearning procedure that initializes a new random model removes all information about $D_f$, satisfying the forgetting criteria. However, all other information is lost alongside. Once unlearning procedures delete $D_f$ factorily, utility determines the value of the model produced. Utility could include desirable properties like robustness \cite{60} and out-of-distribution generalization \cite{36}, but in this work we focus on accuracy.

**Resource Efficiency:** Consider another naive unlearning procedure which trains a new retrained model $M_r$ from scratch on $D \setminus D_f$ with the same (potentially stochastic) training procedure $T$. This procedure satisfies forgetting by never using $D_f$ and typically has high utility. However, retraining from scratch is computationally inefficient. Any proposed unlearning procedure must be significantly more resource-efficient than retraining to be valuable, primarily in terms of the time required to unlearn.

Our paper focuses on inexact-unlearning and scalability to large deletion sets, which we describe below.

**Inexact-unlearning:** The primary goal of forgetting can often be hard to achieve, especially in deep networks. Hence, inexact-unlearning literature has relaxed the forgetting goal in two ways: not provable \cite{10, 32, 35, 39, 45, 56, 67} and imperfect \cite{10, 31, 32, 35, 39, 51, 65, 68}. Relaxing provability implies unlearning methods do not provide any proven guarantees of information removal. Imperfectness relaxes the forgetting requirement for unlearning methods in order to achieve better utility or compute-efficiency, implying that models should try to remove most information specific to the deletion set. The degree of forgetting can either be measured using empirical tests or approximate guarantees. Our work focuses on empirically measuring the amount of forgetting produced by imperfect unlearning methods.

**Scalability to large deletion sets:** Many prior unlearning methods assume tiny deletion sets \cite{11, 30, 53, 61, 68}. However, we argue practical applications typically require the deletion of larger subsets of data, necessitating that unlearning procedures should scale well to this setting.

\[^{3}\text{Not necessarily the maximum utility possible.}\]
2.1. Against Model Indistinguishability

Most prior work in inexact-unlearning [10, 48, 51, 55, 65, 68] defines unlearning as: A procedure has unlearnt a given deletion set $D_f$ if the distribution $\phi_u$ of models it produces is indistinguishable from the distribution $\phi_r$ produced by retraining-from-scratch using training procedure $T$ on $D \setminus D_f$. There are two major problems with this model indistinguishability definition.

First, indistinguishability of the model distributions $\phi_u$ and $\phi_r$ is not necessary for unlearning. To demonstrate this, we exploit the fact that tweaking the training procedure gives significantly different model distributions [70]. Let us generate another model distribution $\phi'_r$ by using training procedure $T'$ on dataset $D \setminus D_f$. $T'$ can be created by varying multiple critical hyperparameters in $T$, like the number of epochs, regularization method, and the learning rate to ensure that $\phi'_r$ is significantly different from $\phi_r$. Now, both $\phi_r$ and $\phi'_r$ achieve forgetting because they both are unaware of the deletion set. However, $\phi_r$ and $\phi'_r$ are significantly different, which illustrates the existence of multiple model distributions that satisfy the goal of unlearning. Thus, the indistinguishability definition excludes most correct solutions by unnecessarily penalizing them for not matching one particular distribution.

Second, the prohibitive cost of collecting a representative sample of deep networks makes it infeasible to measure the equivalence of model distributions. Most prior work addresses this by measuring similarity of weights [39, 61, 68] or outputs [27–29, 51] between a single model sampled from $\phi_u$ and $\phi_r$. However, measuring similarity between two model instances is not representative of the similarity between the distributions they are drawn from. Further, parallel work [62] illustrates how it is possible to obtain a model identical to the original model $M$ even when not using $D_f$ for training. Thus, even $M$ can be drawn from $\phi_r$, in which case performing no unlearning would pass the similarity-to-retrained-model based criterion.

To summarize: model indistinguishability is neither necessary nor sufficient for defining unlearning in practice.

2.2. Scalability to Larger Deletion Sets

Many real-world scenarios require the removal of a large number of samples. Examples include removing noisy labels [49, 50], deleting poisoned samples [40, 43, 66], deleting data that induces harmful biases [20, 52], and organizations requiring deletion of user data older than some retention period. Even in the context of privacy, a single user might own multiple samples in the dataset. In biometrics like face recognition [64], one user may form an entire class [10]. Moreover, user deletion requests may occur in bursts after certain events of interest, such as revelations of privacy leaks by an organization [5]. Lastly, batching online deletion requests requires less invocations of the unlearning procedure, boosting resource efficiency.

![Figure 2](image-url) Hyperbolic deterioration of efficiency in isolation-based unlearning when scaling to a large number of removed samples. In this work, we analyze $|D_f|$ from 100-4000 where $\mathbb{E}[Y] \sim 1$.

2.2.1 Against Data-Influence Isolation

Among recent work that does not rely on the indistinguishability criterion, data-influence isolation approaches form the majority. They achieve the forgetting objective by simply ensuring no access to the deletion set. Isolation-based strategies change the training process by creating an ensemble [11, 30, 35, 53], each of whose models is trained on different subsets of the dataset. This ensures architecturally [6, 11, 53] or temporally [11, 35] isolating the influence of any sample to a limited part of training, requiring retraining for only the affected parts. Isolation has been used across techniques like Linear Classification [6], Random Forest [13, 54], KNN [1], SVM [16, 63] and DNN [11, 30, 35] by utilizing or creating a sparse influence graph [53]. Data-influence isolation often comes at the cost of utility as each portion becomes a weaker learner [8], especially in deep networks [58]. To overcome the dropping utility, the training and unlearning time may need to be increased, reducing resource efficiency.

We show the poor scalability of data-influence isolation approaches to larger deletion set sizes. Let $P$ be the number of parts obtained with the isolation strategy. We assume the best-case scenario where each sample only influences one part. We make the simplifying assumption that the samples are uniformly distributed across parts, and the probability of a removed sample belonging to any particular portion remains constant ($\frac{1}{P}$). Let $Y$ be the number of affected parts. The probability part $i$ is affected by at least one sample in $D_f$ is $1 - (1 - \frac{1}{P})^{|D_f|}$. Thus by the linearity of expectation:

$$\mathbb{E}[Y] = P \left( 1 - \left( 1 - \frac{1}{P} \right)^{|D_f|} \right)$$  \hspace{1cm} (1)

Figure 2 and Appendix Section D.1 demonstrate that the computation costs of isolation-based strategies scale poorly as the deletion set size increases.
Given our forgetting-oriented definition, we now propose evaluations for reliably measuring forgetting with three important qualities not achieved together in existing literature (as shown in Table 1).

**Does not penalize better utility:** We need evaluation methods to not penalize better utility. Consider removing a random subset of samples and computing error on the deletion set [27, 35, 56], with better unlearning models having a higher error. Such an evaluation can not differentiate between an unlearnt model that generalizes well to the (now unseen) deletion set versus one that does not unlearn at all. Similarly, metrics measuring the model indistinguishability criteria exclude a large set of correct solutions, unnecessarily penalizing even the ones with better utility.

**Comparability:** Metrics should enable choosing between unlearning procedures in the presence of differing training procedures and architectures as such changes are often used to enable better unlearning [11, 27–30, 35]. As mentioned in Table 1, L2-weights cannot be compared if there is any architectural modification. Similarly, re-learn time [27–29] is too sensitive to the training procedure $T$.

**Checks Property Generalization:** Zhang et al. [73] demonstrated that deep networks can overfit on randomly labelled training data, a phenomenon known as memorization. In contrast, certain properties only present in $D_f$ might have introduced effects on the classification of unseen samples [46], which we term as property generalization. Removal of such generalized properties is especially important for debiasing, denoising and depoisoning. However, prior tests such as membership inference attacks [17, 57] only try to determine memorization, with no explicit consideration for property generalization. We empirically show how some unlearning methods may remove memorization but not property generalization in Section 4.2.2.

### Table 1. Comparison of metrics used by inexact-unlearning methods in classification settings analyzed across three desiderata: Does not penalize better utility (DPBU), Comparability (Comp), Checks Property Generalization (CPG).

| Metric                  | Test              | DPBU | Comp | CPG |
|-------------------------|-------------------|------|------|-----|
| Relearn time            | RS [27], pCR [28, 29], FCR [27] | ×    | ×    | ×   |
| L2 weights              | RS [19, 61, 68]   | ×    | ×    | ✓   |
| L1 ConfusionMat         | RS [51]           | ×    | ✓    | ✓   |
| L1 Softmax              | RS [27], pCR [29], FCR [27] | ×    | ✓    | ✓   |
| Error                   | RS [27, 35, 56], pCR [28, 29], FCR [27] | ×    | ✓    | ✓   |
| MIA                     | RS [27, 45], pCR [29], FCR [10, 27], IC (Ours) | ✓    | ✓    | ×   |
| Error                   | RC (Ours), IC (Ours) | ✓    | ✓    | ✓   |
| Forgetting Score        | FCR, IC (Ours)    | ✓    | ✓    | ✓   |

**3.1. Interclass Confusion Test**

Evaluations can be compared based on their ability to detect imperfectness in unlearning for a given deletion set size. Black-box evaluations can leverage three strategies to enhance forgetting measurements: (i) Manipulating input data or labels, (ii) Strategically choosing the deletion set and (iii) Computing metrics on probability outputs of the model. Largely overlooked by prior work, the first two aspects can increase the sensitivity of an evaluation by introducing a differentiating influence specific to $D_f$. We leverage them to propose the Interclass Confusion (IC) test as described in Figure 3:

1. Take a part of the training data $S \subset D$ from two chosen classes (Strategic sampling).
2. Swap labels between the two classes for all samples in $S$ (Label manipulation) to get the confused set $S'$. $S'$ and $D \setminus S$ together form the new training data $D'$.
3. Mark the entirety of $S'$ as data to be deleted ($D_f$) from the trained model.
4. Compute metrics using unlearnt model outputs to measure forgetting (Leverage model outputs).

We empirically show that the IC test consistently amplifies the detection of retained information over multiple unlearning metrics.

### 3.1.1 Metrics

Let the confusion matrix generated by model $M$ on any given dataset $S$ be denoted by $C^{M,S}$. Let the confused classes in the IC test be $A$, $B$ and their samples in $S$ be $S_A$ and $S_B$.

Computing error is a standard label-only metric used in prior work which can be adapted to the IC test:

**Error (Err):** It measures the percentage of mistakes made on samples from the affected classes. For IC-test it only considers samples from the confused classes:

$$\text{Err}(M, S, A, B) = \frac{\sum k \cdot C_{A,k}^{M,S} + \sum k' \cdot C_{B,k'}^{M,S}}{|S_A| + |S_B|}, k \neq A, k' \neq B$$

(2)

We further refine (as illustrated in Figure 3) error to give a targeted measurement of the confusion between the two classes by introducing the forgetting score:

**Forgetting Score (Fgt):** It measures the number of samples classified in a way that wouldn’t occur if the information had been fully removed. For the IC-test, it is the number of samples still confused between the two classes:

$$\text{Fgt}(M, S, A, B) = C_{A,B}^{M,S} + C_{B,A}^{M,S}$$

(3)

A lower Fgt is strictly better, with the original model and retrained model simply used for reference. For confusion between $N > 2$ classes, Fgt is the sum of the confusion matrix terms for all pair-wise misclassifications among the $N$ classes. Forgetting score converges to error when $N$ is the same as the total number of classes. Note that better utility would lead to lower Fgt, but the influence of utility is significantly lesser than in Err.

The forgetting score and error when computed on the deletion set $D_f$ measure memorization. On the subset $D_t$ of test set samples from the affected classes, they measure property generalization.

**Confidence-based MIA (CoMI):** Membership Inference Attacks (MIA) [57] aim to infer the presence of particular samples in the training data. Recently, a simple confidence-based MIA (CoMI) [59] has been shown to match the performance of even extensively trained white-box MIA classifiers (see [38] for a survey). For unlearning, we formulate MIA to directly discriminate between outputs on samples from $D_f$ and $D_t$, in contrast, to train and test set used in past literature [27, 29, 30, 45]. We adapt CoMI for the IC test by using the confidence of the confused class labels instead of true classes. A detailed description of CoMI is included in Appendix Section B. As in existing MIA literature, the attack accuracy ranges from 50% (unlearnt) to 100%, with a higher value signifying more confidence in detecting whether a sample was in the training data.

#### 3.1.2 Ablating Targeted Selection of Classes

To ablate the effect of strategic data selection in the IC test, we introduce a random confusion (RC) test. In the RC test, $k$ samples of each class are randomly assigned to any of the other classes. Essentially, the RC test selects samples, i.i.d for mislabelling in contrast to the IC test, which does it with two classes, a strategic non-i.i.d selection. Note that for the RC test, forgetting score is equivalent to test error since the forgetting metric relies on strategic sampling.

### 3.2. Investigating Information Retention in DNNs

We propose a way to determine how early information specific to $D_f$ is encoded in a deep network. For any model $M$, we can obtain an unlearnt model $M_u$ by erasing all information gained from $D_f$ in the final $k$ layers. If the IC test shows $M_u$ still didn’t forget all information, we know some information has to be present in the earlier layers.

**Exact-unlearning the last $k$ layers (EU-$k$):** We freeze the initial layers of the model $M$, and retrain the last $k$ layers from scratch using the same training procedure $T$ on retain set $D \setminus D_f$. Varying $k$ allows us to check how early in the network the test can detect the presence of information.

**Catastrophically forgetting the last $k$ layers (CF-$k$):** Neural Networks suffer from catastrophic-forgetting [24], when a model is continually updated without some previously learnt samples, the model loses knowledge about them. We freeze the initial layers of the model $M$ and finetune the last $k$ layers on the retain set $D \setminus D_f$. We use the same training procedure $T$ but for fewer epochs as the information learnt by the original model is leveraged, making it significantly more efficient than EU-$k$. We measure how close catastrophic-forgetting gets to the complete removal of information about $D_f$ for varying $k$.

Both EU-$k$ and CF-$k$ exploit the fact that a large fraction of computation occurs during convolutions in the early layers [12]. The features produced by the earlier layers can be precomputed once, requiring increasingly less time for unlearning as $k$ decreases. We expect a small $k$ to suffice for mostly forgetting as the earlier layers encode low-level features [71]. We empirically show varying $k$ trades-off forgetting with resource-efficiency at similar utility as $M_r$, thus making EU-$k$ and CF-$k$ valuable baselines for inexact-unlearning.

### 4. Experiments

We now empirically compare our IC test and CF/EU baselines, using them to obtain broader insights. We discuss implementation details in Appendix Section C.
4.1. Experimental Setup

Tests: We compare multiple evaluation methods in our experiments. For a fair comparison, we use the same deletion set size \( n \). The tests are of the following types:

- **Random Samples (RS):** Removing \( n \) random samples from \( D \), and measuring Err and CoMI. It is untargeted and has no labels.
- **Class Removal (CR):** Removing \( n \) samples from a particular class, and measuring Err, CoMI, and Fgt\(^4\). We refer to it as full class removal (FCR) when all samples of a class are removed and partial class removal (pCR) otherwise. It is targeted and has no labels.
- **Random Confusion (RC):** Mislabelling \( n \) samples from \( D \) by assigning them to other randomly chosen classes and measuring Err and CoMI. It is untargeted and has no labels.
- **Interclass Confusion (IC):** Mislabelling \( \frac{n}{2} \) samples from two classes into each other and measuring Err, CoMI and Fgt. It is targeted and has no labels.

Experiments in this section use \( n \) corresponding to the number of training samples in one class: 4000 for CIFAR10 and 400 for CIFAR100 \([42]\). Hence, the CR test completely removes any information about the class. The IC test completely confuses the two classes involved, and in this case robustness to label noise \([47]\) cannot help unlike in RC. We show the effect of varying deletion set size in Appendix Sections E.4 and E.3 and more details about the tests in D.3.

To demonstrate property generalization for the targeted CR and IC tests, metrics are computed only on unseen samples from the involved classes, unlike the untargeted RS and RC test where the full test set is used.

Unlearning Procedures: We empirically benchmark four unlearning methods along with the original \((M)\) and retrain \((M_r)\) models. We compare our methods, exact-unlearning the final \( k \) layers \((EU-k)\) and catastrophic-forgetting on the final \( k \) layers \((CF-k)\), with approaches which can scale to large deletion sets and have made their code available: Fisher \([28]\) and NTK+Fisher (NTK-F) \([29]\).

4.2. Results

We compare various evaluation strategies, unlearning methods and demonstrate the effect of the training procedure on unlearning. All results are averaged over three runs with different random seeds.

4.2.1 Metric Comparisons

The usefulness of a test is established by how reliably it quantifies the remaining information after applying any inexact-unlearning method. We compare existing evaluation methods which use random sampling \([27, 35, 39, 45, 51, 56, 61, 68]\).

\(^4\)For CR test, Fgt calculates the number of samples labelled as the removed class, further details in Appendix Section D.2.2.

| Method | RS-Err \(\uparrow\) | CR-Err \(\uparrow\) | RC-Err \(\downarrow\) | IC-Err \(\downarrow\) | CR-Fgt \(\downarrow\) | IC-Fgt \(\downarrow\) |
|--------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Original | 8.4 ± 0.2 | 3.4 ± 0.4 | 143.3 ± 0.7 | 53.2 ± 1.0 | 1000 ± 19.2 | 901.3 ± 27.1 |
| 1-layer | CF | 8.4 ± 0.1 | 68.1 ± 3.4 | 130.0 ± 0.9 | 409.0 ± 0.4 | 319.0 ± 7.3 | 621.0 ± 21.0 |
| EU | 8.5 ± 0.1 | 100.0 ± 0.0 | 128.5 ± 0.9 | 404.4 ± 0.4 | 0.0 ± 0.0 | 624.5 ± 31.0 |
| 10-layers | CF | 8.4 ± 0.2 | 97.4 ± 1.1 | 120.0 ± 0.6 | 273.0 ± 0.6 | 26.3 ± 0.4 | 349.7 ± 13.1 |
| EU | 8.7 ± 0.1 | 100.0 ± 0.0 | 120.0 ± 0.7 | 281.0 ± 1.8 | 0.0 ± 0.0 | 554.0 ± 13.1 |
| 50-layers | CF | 8.5 ± 0.1 | 100.0 ± 0.0 | 100.0 ± 0.4 | 240.0 ± 8.0 | 0.0 ± 0.0 | 229.0 ± 14.0 |
| EU | 9.1 ± 0.3 | 100.0 ± 0.0 | 104.4 ± 0.4 | 233.3 ± 0.3 | 0.0 ± 0.0 | 209.7 ± 15.0 |
| Retrain | 9.3 ± 0.1 | 100.0 ± 0.0 | 88.0 ± 3.3 | 223.0 ± 0.9 | 0.0 ± 0.0 | 179.7 ± 4.5 |

\(\uparrow\) and \(\downarrow\): In Table 3 we compare different evaluation methods\(^5\) and state consistent observations. Primarily, the IC test is the only test which reliably determines that exactly unlearning the final layer \((1\text{-layer-EU})\) does not remove generalized properties. Neither addition of confusion \((RS \rightarrow CR)\) nor strategic sample selection \((RS \rightarrow CR)\) detects imperfect forgetting of generalized properties alone, both components \((forming \text{the IC-test})\) are necessary. Given EU-50 and the Retrain model are very close on the IC-test, our IC-test can investigate deletion of generalized properties up to almost half the depth of the network. Comparing CF and EU procedures for each value of \(k\), we observe that catastrophic-forgetting is able to unlearn nearly as well as exact-unlearning.

Memorization (Measured on \(D_{f}\)): In Table 3 we compare different evaluation methods and state consistent observations. CR test is not able to detect memorized information even in exact-unlearning of the final layer. IC is the only test that clearly demonstrates the forgetting trends across unlearning more layers when using CoMI as the metric. In contrast, RS-CoMI gives a weak signal while CR-CoMI and RC-CoMI do not give any insights. On Err metric, RS seems to give clear trends, but its interpretation is unclear since RS-Err penalizes better utility. Specifically, a perfectly unlearned model can give a low RS-Err if it simply generalizes to the now unseen deletion set. In contrast, both confusion-based tests are free from such issues and show the degree of memorization across layers. Comparing CF and EU procedures for each value of \(k\), we observe that catastrophic-forgetting is...
is slightly worse than exact-unlearning.

Overall, the IC-test is the most effective in distinguishing between varying forgetting efficacy across metrics. For a given k, catastrophic-forgetting removes almost as much information gained from \( D_f \) as exact-unlearning while being twice as resource-efficient. EU-1 represents an upper limit on the forgetting efficacy of applying unlearning on the final linear layer. We find that even EU-1 leaks a lot of information gained from \( D_f \), severely limiting the usefulness of a series of prior works [10,39,68] for deep networks.

### 4.2.2 Unlearning Method Comparisons

**Setup:** We use Small-CIFAR-5, a subset of CIFAR10, as in Golatkar et al. [28,29] for comparing Fisher and NTK-F as they cannot scale to larger datasets like CIFAR10 because they require extensive memory. Both Fisher and NTK-F require a variety of training assumptions not found in standard training procedures. We use the training procedure of Golatkar et al. to benchmark Fisher and NTK-F while following our standard training procedure for our unlearning methods. We present results on both the CR and IC test in Table 4.

**Utility:** Our original and retrained models have a significantly lower test set error and hence better utility. However, they overfit more to the train set resulting in more memorization.

**Forgetting:** Surprisingly, Fisher does worse than no unlearning in both evaluation settings, across all metrics (Fgt, Err, Test Error, Time). EU-1 outperforms CF-10 in both tests, indicating catastrophic-forgetting is less effective on tiny datasets. On the CR test, NTK-F seems to achieve only

---

Table 3. Comparison of evaluation methods for memorization. Scores are reported as: mean ± stdev. The IC-test amplifies memorization removal trends across metrics.

| Model      | IC-Fgt (\( \downarrow \)) | Err (\( \downarrow \)) | Test Err (\( \downarrow \)) | Time(s) (\( \downarrow \)) |
|------------|--------------------------|-----------------------|----------------------------|--------------------------|
|            | T from [28,29]           |                       |                            |                          |
| Original   | 92.3 ±121.6              | 8.3 ±70               | 26.7 ±0                    | 0.0 ±0                   |
| Fisher     | 94.6 ±157.0              | 6.0 ±56               | 33.2 ±0                    | 141 ±91                 |
| NTK-F      | 77.0 ±119.6              | 23.0 ±20.3            | 31.0 ±0                    | 141 ±90                 |
| Retrain    | 0.0 ±0                   | 100.0 ±100.0          | 41.4 ±0                    | 9.81 ±0                  |

Table 4. Comparison between methods on FCR-Test (left) and IC-Test (right) on Small-CIFAR-5. Scores are reported as: Memorization | Property Generalization for Fgt and Err, along with the error on full test set and unlearning time.

| Model      | IC-Fgt (\( \downarrow \)) | Err (\( \downarrow \)) | Test Err (\( \downarrow \)) | Time(s) (\( \downarrow \)) |
|------------|--------------------------|-----------------------|----------------------------|--------------------------|
|            | T from [28,29]           |                       |                            |                          |
| Original   | 54.3 ±85.0               | 65.5 ±58.6            | 42.8 ±0                    | 0.0 ±0                   |
| Fisher     | 57.6 ±96.6               | 64.3 ±56.0            | 43.7 ±0                    | 142 ±73                 |
| NTK-F      | 9.6 ±31.0                | 18.6 ±25.1            | 31.1 ±0                    | 142 ±76                 |
| Retrain    | 9.0 ±13.3                | 25.0 ±18.3            | 30.5 ±0                    | 9.15 ±0                  |

---
marginal forgetting, performing dramatically worse than even our final layer unlearning methods. On the IC-Test, NTK-F is able to achieve a reduction in memorization comparable to a retrained model. However, NTK-F does poorly in removing property generalization, worse than even CF-1 and EU-1, highlighting the need for a separate analysis of property generalization not found in existing literature.

Fisher and NTK-F require compute costs for obtaining the Fisher Information Matrix, to the extent that they take far more unlearning time than even retraining from scratch. Overall, our exact-unlearning the final layer baseline achieves better forgetting, utility, and resource efficiency than previously proposed unlearning methods that are applicable to deep networks and scalable to large deletion sizes.

### 4.2.3 Amenability of Original Model to Unlearning

Intuitively, different original models can show varying propensities to memorize $D_f$, and some may be able to remove more information than others even using the same unlearning procedures. We investigate the effects of changing the training procedure $T$ when training the original model $M$ on the efficacy of unlearning. We found regularization methods, like cutmix [72] and early stopping, had the most prominent effects on forgetting while providing better utility and resource-efficiency respectively.

From Table 5, our primary observation is that both early stopping and cutmix greatly reduce memorization of confused labels in the original model. Unlearning 10-layers from a regularized model achieves a much greater decrease in both memorization and property generalization. This indicates better regularization of the original model makes it easier to remove information; however, as better unlearning deletes more information, the benefits of regularization on unlearning decrease.

Overall, the training procedure for the original model can have a significant impact on the efficacy of unlearning. Thus, when no training assumptions are involved, a fair comparison of unlearning methods requires starting from similarly trained original models.

### 4.3. Limitations

Despite being a far more effective method to measure the efficacy of unlearning than past evaluation strategies, the IC test has some limitations. Primarily, label confusion is a very specific way to measure forgetting and unlearning procedures may adapt to it. Thus, passing the IC-test alone cannot guarantee forgetting. Second, our evaluation does not completely disentangle forgetting from utility. In particular, increasing model accuracy can improve the forgetting score without removing more information specific to $D_f$. Even on completely removal of information about $D_f$, the forgetting score can be non-zero if some samples are mislabelled between the confused classes. It is thus not clear when an unlearning method passes our evaluation. One can only claim an unlearning procedure to be “better” or “worse” based on how low the forgetting score is. Finally, our evaluation produces aggregates scores over a deletion set; whether any individual sample was forgotten is not obtained.

### 5. Conclusion

In this work, we highlighted important shortcomings of popular paradigms of unlearning. We motivated the need to unlearn large deletion sets, to which data-influence isolation approaches scale poorly. To overcome the limitations of indistinguishability-based definitions, we redefined the goal of unlearning by splitting it into forgetting, utility and resource-efficiency. We emphasised how forgetting involves removing two types of information gained specifically from data to be deleted: memorization and property generalization. Our black-box Interclass Confusion test is domain-agnostic as it only affects labels. It is effective in creating a wider spectrum for unlearning, providing reliable evaluations across metrics. We devised unlearning baselines to investigate whether information specific to the deletion set is present in the early layers of the network.

Overall, we hope that our definition of unlearning and IC test will guide the design and evaluation of future approaches for inexact-unlearning. Our work raises various future directions to be investigated: (1) How do we formalize the notion of additional information contributed by $D_f$? (2) Is it possible to devise a metric that completely disentangles forgetting from utility? (3) Can we better investigate where information about any subset of samples is stored in a deep network to allow targeted deletion?

**Acknowledgements:** We thank Shyamgopal Karthik, Saujas Vaduguru, Arjun T.H., Naren Akash RJ, Shradha Se-
hgal, Nikhil Chandak, Shashwat Chandra among others for helpful feedback.

References

[1] Alternative k-nearest neighbour rules in supervised pattern recognition: Part 1. k-nearest neighbour classification by using alternative voting rules. *Analytica Chimica Acta*, 1982. 3

[2] California consumer privacy act, 2018. https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=201720180AB375. 1

[3] Humans forget, machines remember: Artificial intelligence and the right to be forgotten. *Computer Law & Security Review*, 2018. 1

[4] Personal information protection and electronic documents act, 2018. https://www.priv.gc.ca/en/opc-news/news-and-announcements/2018/an_181010v1. 1

[5] Alessandro Acquisti, Allan Friedman, and Rahul Telang. Is there a cost to privacy breaches? an event study. 2006. 3

[6] Nasser Aldaghi, Hessam Mahdavifar, and Ahmad Beirami. Coded machine unlearning. In *IEEE Access*, 2021. 3

[7] Devansh Arpit, Stanislaw Jastrz̆bowski, Nicolas Balas, 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 *ICML*, 2017. 2

[8] Michele Banko and Eric Brill. Scaling to very very large corpora for natural language disambiguation. ACL, 2001. 3

[9] Marco Barreno, Blaine Nelson, Russell Sears, Anthony D. Joseph, and J Doug Tygar. Can machine learning be secure? In *ASIA CCS*, 2006. 1

[10] Thomas Baumhauer, Pascal Schöttle, and Matthias Zeppelzauer. Machine unlearning: Linear filtration for logit-based classifiers. arXiv preprint arXiv:2002.02730, 2020. 1, 2, 3, 4, 6, 7, 11, 13

[11] Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *IEEE S&P*, 2021. 1, 2, 3, 4

[12] Andrew Brock, Theodore Lim, J. M. Ritchie, and Nick Weston. Freezeout: Accelerate training by progressively freezing layers, 2017. 5

[13] Jonathan Brophy and Daniel Lowd. Machine unlearning for random forests. In *ICML*, 2021. 3

[14] Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *IEEE S&P*, 2015. 1

[15] Nicholas Carlini, Chang Liu, Ulfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *USENIX Security*, 2019. 1

[16] Gert Cauwenberghs and Tomaso Poggio. Incremental and decremental support vector machine learning. In *NeurIPS*, 2000. 3

[17] Min Chen, Zhikun Zhang, Tianhao Wang, Michael Backes, Mathias Humbert, and Yang Zhang. When machine unlearning jeopardizes privacy. In *CCS*, 2021. 4

[18] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017. 1

[19] Council of European Union. Council regulation (EU) no 269/2014, 2014. http://eur-lex.europa.eu/legal-content/EN/TXT/?qid=1416170084502&uri=CELEX:32014R0269. 1

[20] Matteo Fabbrizzi, Symeon Papadopoulos, Eirini Ntoutsi, and Ioannis Kompatsiaris. A survey on bias in visual datasets. *arXiv preprint arXiv:2107.07919*, 2021. 1, 3

[21] Federal Trade Commission. California company settles FTC allegations it deceived consumers about use of facial recognition in photo storage app, January 2021. 1

[22] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confusion information and basic countermeasures. In *CCS*, 2015. 13

[23] Benoit Frey and Michel Verleysen. Classification in the presence of label noise: A survey. In *IEEE NNLS*, 2014. 1

[24] Robert M. French. Catastrophic forgetting in connectionist networks. In *Trends in Cognitive Sciences*, 1999. 2, 5

[25] Shaoqiang Fu, Fengxiang He, Yue Xu, and Dacheng Tao. Bayesian inference forgetting, 2021. 1

[26] Antonio Ginart, Melody Y. Guan, Gregory Valiant, and James Zou. Making AI forget you: Data deletion in machine learning. In *NeurIPS*, 2019. 1

[27] Aditya Golatkar, Alessandro Achille, Awinash Ravichandran, Marzia Polito, and Stefano Soatto. Mixed-privacy forgetting in deep networks. In *CVPR*, 2021. 1, 3, 4, 5, 6, 11, 13

[28] Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *CVPR*, 2020. 1, 3, 4, 6, 7, 13

[29] Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Forgetting outside the box: Scrubbing deep networks of information accessible from input-output observations. In *ECCV*, 2020. 1, 3, 4, 5, 6, 7, 11, 13

[30] Laura Graves, Vineel Nagisetty, and Vijay Ganesh. Amnesiac machine learning. In *AAAI*, 2021. 2, 3, 4, 5, 11, 12, 13

[31] Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens van der Maaten. Certified data removal from machine learning models. In *ICML*, 2020. 1, 2

[32] Tomohiro Hayase, Suguru Yasutomi, and Takashi Katoh. Selective forgetting of deep networks at a finer level than samples. In *AAAI-W*, 2021. 2

[33] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2015. 12

[34] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 12

[35] Yingzhe He, Guozhu Meng, Kai Chen, Jinwen He, and Xingbo Hu. Deep oblivious: A powerful charm for erasing data residual memory in deep neural networks. *arXiv preprint arXiv:2105.06209*, 2021. 2, 3, 4, 6

[36] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *ICLR*, 2017. 2
Sara Hooker. Moving beyond “algorithmic bias is a data problem”. Patterns, 2021. 1

Hongsheng Hu, Zoran Salcic, Gillian Dobbie, and Xuyun Zhang. Membership inference attacks on machine learning: A survey. arXiv preprint arXiv:2103.07853, 2021. 5

Zachary Izzo, Mary Anne Smart, Kamalika Chaudhuri, and James Zou. Approximate data deletion from machine learning models. In AISTATS, 2021. 2, 3, 4, 6, 7

Matthew Jagielski, Alina Oprea, Battista Biggio, Chang Liu, Cristina Nita-Rotaru, and Bo Li. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In IEEE S&P, 2018. 3

Shari Theresa Jose and Osvaldo Simeone. A unified pac-bayesian framework for machine learning via information risk minimization. In ICML, 2021. 1

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 6

Yiming Li, Baoyuan Wu, Yong Jiang, Zhifei Li, and Shu-Tao Xia. Backdoor learning: A survey. arXiv preprint arXiv:2007.08745, 2020. 3

Bo Liu, Ming Ding, Sina Shaham, Wenny Rahayu, Farhad Farokhi, and Zihuai Lin. When machine learning meets privacy: A survey and outlook. In ACM Computing Surveys, 2021. 1

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. 2, 4, 5, 6, 11, 12

Preetum Nakkar and Yamini Bansal. Distributional generalization: A new kind of generalization, 2020. 2, 4

Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. NeurIPS, 2013. 6

Seth Neel, Aaron Roth, and Saeed Sharifi-Malvajerdi. Descent-to-delete: Gradient-based methods for machine unlearning. In COLT, 2021. 1, 3

Curtis G. Northcutt, Anish Athalye, and Jonas Mueller. Persasive label errors in test sets destabilize machine learning benchmarks, 2021. 1, 3

Curtis G. Northcutt, Lu Jiang, and Isaac L. Chuang. Confident learning: Estimating uncertainty in dataset labels. In JAIR, 2021. 1, 3

Alexandra Peste, Dan Alistarh, and Christoph H Lampert. Ssse: Efficiently erasing samples from trained machine learning models. arXiv preprint arXiv:2107.03860, 2021. 1, 2, 3, 4, 6

Vinay Uday Prabhu and Abeha Birhane. Large image datasets: A pyrrhic win for computer vision? In WACV, 2021. 1, 3

Sebastian Schelter. “amnesia” - machine learning models that can forget user data very fast. In CIDR, 2020. 2, 3

Sebastian Schelter, Stefan Grafterger, and Ted Dunning. Hedgecut: Maintaining randomised trees for low-latency machine unlearning. In SIGMOD/PODS, 2021. 3

Ayush Sekhari, Jayadev Acharya, Gautam Kamath, and Ananda Theertha Suresh. Remember what you want to forget: Algorithms for machine unlearning. In NeurIPS, 2021. 3

Takashi Shibata, Go Irie, Daiki Ikami, and Yu Mitsuzumi. Learning with selective forgetting. In IJCAI, 2021. 2, 4, 6

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In IEEE S&P, 2017. 1, 4, 5, 11

Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. Journal of Big Data, 2019. 3

Liwei Song and Prateek Mittal. Systematic evaluation of privacy risks of machine learning models. In USENIX, 2021. 5, 11, 12

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199, 2013. 2

Anvith Thudi, Gabriel Deza, Varun Chandrasekaran, and Nicolas Papernot. Unrolling sgd: Understanding factors influencing machine unlearning. arXiv preprint arXiv:2109.13398, 2021. 2, 3, 4, 6

Anvith Thudi, Hengrui Jia, Iilia Shumailov, and Nicolas Papernot. On the necessity of auditable algorithmic definitions for machine unlearning. arXiv preprint arXiv:2110.11891, 2021. 3

Cheng-Hao Tsai, Chieh-Yen Lin, and Chih-Jen Lin. Incremental and decremental training for linear classification. In KDD, 2014. 3

Matthew Turk and Alex Pentland. Eigenfaces for recognition. Journal of cognitive neuroscience, 1991. 3

Enayat Ullah, Tung Mai, Anup Rao, Ryan Rossi, and Raman Arora. Machine unlearning via algorithmic stability. In COLT, 2021. 1, 2, 3

Bolun Wang, Yuanshun Yao, Shawn Shan, Huiyi Li, Bimal Viswanath, Haitao Zheng, and Ben Y. Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In IEEE S&P, 2019. 3

Alexander Warnecke, Lukas Pirch, Christian Wressnegger, and Konrad Rieck. Machine unlearning of features and labels. arXiv preprint arXiv:2108.11577, 2021. 2, 12

Yinjun Wu, Edgar Dobriban, and Susan B. Davidson. DeltaGrad: Rapid retraining of machine learning models. In ICML, 2020. 1, 2, 3, 4, 6, 7

Chaofei Yang, Qing Wu, Hai Li, and Yiran Chen. Generative poisoning attack method against neural networks. In Information & Security, 2020. 1

Li Yang and Abdallah Shami. On hyperparameter optimization of machine learning algorithms: Theory and practice. Neurocomputing, 2020. 3

Jason Yosinski, Jeff Clune, Thomas Fuchs, and Hod Lipson. Understanding neural networks through deep visualization. In IJCAI-W, 2015. 5

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuik Cho, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In CVPR, 2019. 8

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In ICLR, 2017. 4
### Table 6. Reference for notations used in this work

| Abbr. | Definition | Ref. in Section |
|-------|------------|-----------------|
| IC    | Interclass Confusion test | 3.1, 4.1. |
| RC    | Random Confusion test | 3.1.2, 4.1. |
| RS    | Random Samples test | 4.1. |
| CR    | Class Removal test | 4.1. |
| pCR   | Partial Class Removal test | 4.1. |
| FCR   | Full Class Removal test | 4.1. |
| Fgt   | Forgetting Score metric | 3.1.1. |
| Err   | Error metric | 3.1.1 |
| MIA   | Membership Inference Attack(s) | 3.1.1, App. Sec. 2. |
| CoMI  | Confidence-based Membership Inference attack | 3.1.1, App. Sec. 2. |
| CF-k  | Catastrophic-Forgetting $k$ layers | 3.2. |
| EU-k  | Exact-Unlearning $k$ layers | 3.2. |
| DNN   | Deep Neural Network | |
| i.i.d | Independent and identically distributed | |
| NTK-F | Neural Tangent Kernel + Fisher method | |
| stdev | Standard Deviation | |

### Table 7. Reference for abbreviations used in this work

| A. Reference: Abbreviations |
|----------------------------|

In Tables 6 and 7 we list the terms and notations used in this work respectively. The reader is encouraged to use these for reference.

### B. Membership Inference Attacks

#### B.1. Background

Membership Inference Attacks (MIA) [57] can be used to determine whether a particular sample was part of the training data of a model. Many different black-box formulations of MIA have been used to measure the efficacy of unlearning. Most [27, 29, 30, 45] learn a binary attack classifier: based on the model’s output for the sample, was the sample in the seen training set (class 0) or the unseen test set (class 1)? The attack classifier is then applied on deletion set samples, with ideal unlearning entailing all samples are classified as unseen. However, such a test is extremely sensitive to the efficacy of the attack classifier which may be unreliable. Another approach has been to train the attack classifier to distinguish the outputs of a large number of original ($M$) and retrained ($M_r$) models and then classify the unlearnt model $M_u$ [10]. This formulation involves prohibitive computational expense and applies only when using the problematic indistinguishability definition.

Song et al. [59] recently show that metric-MIA, measuring simple metrics and deciding membership based on a threshold, can match the classification accuracy of trained attack models. In particular, their confidence-based MIA measures the model’s output probability for the target class and selecting separate class-wise membership thresholds. It
is shown to match the performance of even white-box MIA attack classifiers.

### B.2. Our Formulation

We adapt the confidence-based MIA [59] to propose an efficient black-box MIA formulation specifically tailored for measuring forgetting. We assume direct access to the actual model outputs instead of shadow models [30, 45], as shadow models only weaken the attack, making the unlearning test artificially easier to pass. We distinguish the model outputs on deletion set samples and unseen samples from the same underlying distribution rather than training an attack classifier using the entire train and test set.

Our MIA takes in model \( M \), forget set \( D_f \) and unseen samples \( D_u \) from the same classes found in \( D_f \). The following procedure is repeated for each ‘target class’ \( t \):

- Dataset \( D_{MIA} \) is created with the probability outputs for class \( t \): \( M(D_f), \) and \( M(D_u) \), stored as class 0 and class 1 respectively.
- We then create a 50-50\(^6\) shadow \( (D_{MIA-S}) \) - test \( (D_{MIA-T}) \) split of \( D_{MIA} \).
- A threshold \( p_t \) needs to be chosen such that probabilities > \( p_t \) are classified as class 0, and probabilities < \( p_t \) as class 1. The \( p_t \) that maximizes the accuracy on \( D_{MIA-S} \) is chosen.
- The accuracy obtained on \( D_{MIA-T} \) using threshold \( p_t \) is the MIA accuracy for target class \( t \). A weighted average of this test accuracy across all target classes is taken as the final MIA accuracy.

Usually, the target class \( t \) is the actual label of the sample. However, in the case of IC test, we use the mislabelled class as the target for both, \( D_f \) and \( D_u \) samples. Intuitively, the memorization of mislabels in the deletion set would make the wrong class probability output unnaturally higher than other unseen samples of the same class, making the MIA stronger. Such an enhancement is not possible in the case of RC as the mislabels are untargeted.

In line with existing MIA literature, we want our attack classifier accuracy to be 50% incase of no classifier advantage. Thus as the forget set and unseen set may have differing sizes in some experiments, we take a random subset of the larger one to make the attack dataset balanced. The numbers reported are averaged over 20 runs with randomness induced by the subset sampling step. Note that since the classifier learns to distinguish between the test and forget set distribution directly, it might be able to distinguish them spuriously, leading to slightly more than 50% attack classifier accuracy even on perfect unlearning. Thus, the reference gold standard MIA performance can instead be that of any exactly unlearnt model upon undergoing the same evaluation.

### C. Implementation Details

We use CIFAR10 and CIFAR100 datasets with a 40,000-10,000-10,000 train-valid-test split.

We use the ResNet architecture [34] with 110 layers. Our standard training procedure \( T \) is the following: We train our models initialized as Kaiming Normal [33] for 62 epochs (CIFAR10) or 126 epochs (CIFAR100), using a SGD optimizer with momentum 0.9 and weight decay 5e-5, an SGD scheduler with \( t_{mult} = 2, t_0 = 1, \) minlr = 5e-3, maxlr = 0.01 and a batch size of 64. For EU-\( k \) and CF-\( k \) we use the same choices described above, but on the final \( k \) layers. In CF-\( k \), we finetune for only half the epochs to obtain a two-fold speedup over EU-\( k \). The choices made specific to the different evaluation methods are highlighted in Sections B.2, D.3.

The setup used for all experiments is a PC with a Intel(R) Xeon(R) E5-2640 2.40 GHz CPU, 128GB RAM and 1 GeForce RTX 2080 GPU.

We make the following deviations for particular experiments:

- In Table 4 we make changes described in Section D.4.
- In Tables 5, 9, ?? we change the training procedure. When using cutmix regularization, we use \( p = 0.5 \) and \( \alpha = 1.0 \). For early stopping, we halve the number of epochs both while training the original/retrain models and also in the unlearning procedures.
- In Table 10 we vary the number of finetuning epochs in CF-\( k \).
- In Table 15 we vary the confused classes in the IC test from easy-hard on the axis of distinguishability.
- In Figure 5 we further benchmark on ResNet-20, ResNet-56 and ResNet-110 to show our results are robust to the choice of network depths.
- In Table 16 we ablate the effect of warm restarts in training the original/retrain model.

### D. Additional Details and Analysis

We now provide some additional details for results shown in the main paper.

#### D.1. Against Isolation Strategies

We also show that the probability of full-retrain in data-influence isolation unlearning methods scales poorly with increasing deletion set size. Let \( p(n) \) be the probability that all \( P \) portions are affected on the removal of \( n \) samples. Extending the analysis of Warnecke et al. [67] from the specific case of SISA to data-influence isolation in general, we get:

\[
p(n) = 1 - \frac{\sum_{j=1}^{k}(-1)^{j+1}\binom{P}{j}(|P|-j)^n}{|P|^n}
\]

Figure 4 shows \( p(|D_f|) \) grows logarithmically, implying there

---

\(^6\)Given that only 1 parameter (threshold) needs to be learnt, the shadow size is sufficient
is a fast increase in the chance of needing a full-retrain as deletion sets get larger. This demonstrates how data-influence isolation provides little improvement in efficiency compared to the retrain-from-scratch baseline for practical scenarios.

### D.2. Metrics

#### D.2.1 On the Metric Comparisons Table

Note that the list of metrics in Table 1 of the main paper does not include metrics like upper bound on information remaining in weights and activations [27–29] since its unclear whether such metrics can be computed on methods other than their own proposed unlearning procedure. We also exclude purely-qualitative tests such as model inversion attacks [22] which have been used in prior unlearning works [10, 30].

#### D.2.2 Details of Metric Computation

For clarity, we further describe the computation of some metrics. CoMI has already been described in Appendix Section B. Note that for measuring memorization, the deletion set is used, while for measuring generalization (a subset of) the test set is used.

**IC-Fgt**: For the IC test between class A and B, the forgetting score represents the number of samples of class A mislabelled as class B and vice-versa. Intuitively, as the mislabelled samples are forgotten by the unlearning procedure, the model should confuse lesser samples between these two classes.

**CR-Fgt**: For the CR test removing samples from class C, the forgetting score represents the number of samples the model classifies as class C. Intuitively, as more samples from C are removed, the model should classify lesser samples into C. Note that if the entire class is not removed, a model that generalizes better from the partial samples still available may get penalized unnecessarily.

**RC, RS-Err**: Error on unseen samples from the confused classes of the IC test, A and B.

**IC-Err**: Error on unseen samples from the confused classes of the IC test, A and B.

**RC, RS-Err**: Error on all samples from the test set. Here, a specific set of classes cannot be used for a targeted measurement. In Table 8 we show the utilities of the EU-k and CF-k unlearning procedures across all four tests. We observe a negligible impact on utility compared to retraining from scratch, unless utility is correlated with unlearning in the applied test.}

| Method | RS (↓) | CR (↓) | RC (↓) | IC (↓) |
|--------|--------|--------|--------|--------|
| CIFAR-10 (|Df| = 4000) | | | | |
| Original | 8.4 ± 0.2 | 8.8 ± 0.4 | 14.3 ± 0.7 | 6.9 ± 0.5 |
| 1-layer CF | 8.4 ± 0.1 | 8.3 ± 0.2 | 13.0 ± 0.9 | 6.5 ± 0.5 |
| EU | 8.5 ± 0.1 | 8.4 ± 0.3 | 12.8 ± 0.9 | 6.6 ± 0.3 |
| 10-layers CF | 8.4 ± 0.2 | 8.4 ± 0.2 | 12.0 ± 0.6 | 6.5 ± 0.4 |
| EU | 8.7 ± 0.1 | 8.6 ± 0.2 | 12.0 ± 0.7 | 6.7 ± 0.2 |
| 50-layers CF | 8.5 ± 0.1 | 8.1 ± 0.4 | 10.0 ± 0.4 | 6.1 ± 0.3 |
| EU | 9.3 ± 0.3 | 8.8 ± 0.4 | 10.4 ± 0.4 | 6.9 ± 0.5 |
| Retrain | 9.3 ± 0.1 | 8.2 ± 0.3 | 8.8 ± 0.3 | 6.4 ± 0.2 |
| CIFAR-100 (|Df| = 400) | | | | |
| Original | 32.1 ± 1.1 | 31.6 ± 1.1 | 32.4 ± 1.4 | 31.8 ± 0.8 |
| 1-layer CF | 32.1 ± 1.0 | 31.7 ± 1.2 | 32.4 ± 1.3 | 31.7 ± 0.7 |
| EU | 32.1 ± 1.0 | 31.7 ± 1.1 | 32.4 ± 1.2 | 31.8 ± 0.7 |
| 10-layers CF | 32.4 ± 0.9 | 32.2 ± 1.1 | 32.6 ± 1.3 | 32.0 ± 0.6 |
| EU | 33.3 ± 1.2 | 32.5 ± 1.1 | 33.3 ± 1.2 | 32.8 ± 0.9 |
| 50-layers CF | 31.7 ± 0.9 | 31.5 ± 1.0 | 31.8 ± 0.8 | 31.2 ± 0.6 |
| EU | 32.1 ± 0.3 | 31.6 ± 0.2 | 31.8 ± 1.0 | 31.7 ± 1.0 |
| Retrain | 32.2 ± 0.4 | 31.6 ± 0.6 | 31.7 ± 1.3 | 31.8 ± 1.0 |

Table 8. Error on the retain set distribution of test samples across unlearning tests. Scores are reported as: mean ± stdev. The EU-k and CF-k unlearning procedures lead to a minimal change in utility compared to retraining from scratch, unless utility is correlated with unlearning in the applied test.
Method | None | Early Stop | Cutmix | Cutmix+Early
--- | --- | --- | --- | ---
CIFAR-10 (|$|D_f| = 4000$)
Original | 6.53 | 7.91 | 5.81 | 8.02
10-layers CF | 6.11 | 7.93 | 5.45 | 7.27
EU | 6.55 | 7.88 | 5.68 | 7.08
50-layers CF | 5.75 | 7.31 | 5.32 | 6.75
EU | 6.57 | 7.87 | 6.16 | 8.20
Retrain | 6.31 | 8.50 | 5.70 | 8.97
CIFAR-100 (|$|D_f| = 400$)
Original | 32.53 | 33.10 | 27.26 | 30.03
10-layers CF | 32.22 | 33.04 | 27.98 | 30.61
EU | 33.25 | 33.23 | 28.59 | 31.23
50-layers CF | 30.93 | 32.37 | 27.92 | 29.37
EU | 31.98 | 33.66 | 30.41 | 30.93
Retrain | 30.64 | 32.62 | 26.67 | 30.67

Table 9. Error on the retain set distribution of test samples on varying the training procedure of the original model. Regularized models have better utility even after unlearning.

deletion set size is the same as the number of samples from one class in the training set. Note that while the size of $D_f$ is the same when comparing different tests, the size of $D_t$ is dependent on the test itself. In targeted tests (CR, IC), $D_t$ only has test set samples from the affected classes, whereas in untargeted tests (RS, RC) $D_t$ consists of the entire test set. In CR test we remove class 0 for both CIFAR10 and CIFAR100, whereas in RS and RC we draw an equal number of samples randomly from each class.

D.4. Details of Comparison with Golatkar et al.

For comparisons with Golatkar et al., we use their code, dataset and training procedure. The only change we make is using the ResNet-20 architecture as is standard for CIFAR10 to obtain results of both their and our procedures. Golatkar et al. benchmark on a subset of 500 images (100 each from 5 classes) of CIFAR10, which they call Small-CIFAR5. We tried benchmarking their unlearning procedure on models obtained using our training procedure, but possibly due to the violation of some of their training assumptions, we obtained models with near-random performance.

D.5. Utilities

To measure utility, we compute error on unseen samples from the same distribution as $D \setminus D_f$, called the retain distribution. For the RS and RC tests, as the removal is untargeted, the evaluated samples are the same as the full test set. For the CR and RC tests the evaluated samples consist of test set samples from the unaffected classes. This is done as error on samples from the deletion set distribution correlates with the unlearning efficacy, and thus removing them leads to a measurement of utility independent of unlearning.

In Table 9 we show the impact of regularization on utility. We observe that early stopping slightly increases the

errors, while cutmix alone reduces them especially in CIFAR100. Given the significant improvement in utility and greater downstream amenability to unlearning, using regularizers like Cutmix seems highly rewarding. Our unlearning procedures do not decrease the utility barring a slight deterioration when the training procedure uses cutmix while the unlearning procedure does not.

E. Additional Experiments

Finally, we vary some of the choices we make in our experiments to demonstrate the robustness of our observations.

E.1. Varying the number of unlearning epochs

The original experiments train CF-$k$ models for half the epochs of the original training procedure and EU-$k$ models. In Table 10 we compare the variation of performance among CF-$k$ models at the end of each warm restart while finetuning. While less information is unlearnt on reducing epochs, even six epochs are sufficient for drastic improvements in forgetting. We notice that there is no significant change in error on unseen samples. The number of catastrophic forgetting epochs can thus be used to further control the forgetting-efficiency tradeoff at similar utility.
Figure 5. Variation in forgetting efficacy across number of layers unlearnt using CF-$k$ and EU-$k$. The annotations represent (Memorization|Generalization) based on the IC test forgetting score. Most information is present in the final layers of the network, and applying unlearning to the earlier half of the network provides negligible benefits. CF-$k$ consistently performs close to EU-$k$.

E.2. Varying the number of layers

In Figure 5 we show results of varying $k$ for 3 different ResNet depths: 20, 56 and 110. The IC test is able to detect
Table 11. Varying $|D_f|$ for RC test. Results are reported as Memorization|Property Generalization. Error reliably measures memorization even in small deletion sets, though larger ones are needed to produce detectable effects on property generalization.

| Method | Err (%) | CoMI (%) |
|--------|---------|----------|
|        | 400     | 2k       |
|        | 8k      | 20k      |
|        | 400     | 2k       |
|        | 8k      | 20k      |

Table 12. Varying $|D_f|$ for RS test. Results are reported as Memorization|Property Generalization. Error seems to distinguish varying levels of unlearning, but needs huge deletion sets in the case of property generalization. Moreover, here error has the limitation of misaligning forgetting ($\uparrow$) and utility ($\downarrow$).

| Method | Err (%) | CoMI (%) |
|--------|---------|----------|
|        | 400     | 2k       |
|        | 8k      | 20k      |
|        | 400     | 2k       |
|        | 8k      | 20k      |

retained information despite exact unlearning of 30% of the final layers. However, on unlearning the final half of the network, its unclear whether most information is removed or the IC test is unable to identify the presence of retained information. CF-k is consistently within a small margin of EU-k demonstrating the catastrophic forgetting is able to lose enough information to match EU while being two times faster.

E.3. Varying Amount of Untargeted Removal

In Tables 12 and 11 we show the forgetting performance when we vary deletion set sizes in tests with untargeted removal, RS and RC. Here, we use larger sizes than those reported in the main paper as smaller deletion sets show negligible trends in untargeted removal. For detecting effects on property generalization, Err on RC test needs far fewer samples than Err on RS. For memorization, we see that Err is able to distinguish and rank models fairly well whereas CoMI works well in the case of RS test but fails completely on the RC test. CF models continue to be close to EU models here and the gap between them decreases as we add more confusion. Overall, untargeted removal requires much larger deletion sets to show clear forgetting trends as compared to targeted removal, demonstrating the usefulness of strategic sampling.

E.4. Varying Amount of Targeted Removal

Now, we study the forgetting performance for partial Class Removal (pCR) and partial Interclass Confusion. We show results for varying $|D_f|$ from 10% samples of a class to the size of an entire class (as used in the original paper).

First, we present the results of the IC test in Table 13. We see that for memorization (left of $\uparrow$) all metrics are reflective even when a very small subset of samples is confused. Err and CoMI have increasingly better contrast for smaller deletion sets. In contrast, Fgt works consistently well across all deletion set sizes with roughly similar amplitudes of differences. For property generalization (right of $\downarrow$), we have somewhat but not substantially weaker trend in error as compared to Fgt as sample sizes decrease. Both are able to reflect trends in unlearning property generalization.

Then, we present the results of the CR test in Table 14. The CR test has significantly different behavior when all samples of the class are removed (FCR) compared to partial removal (pCR). In the case of FCR, all information about the class is removed, and hence an unlearnt model is expected to not classify any sample as the removed class. However, in pCR a well generalized model may correctly classify more samples than Err on RS. For memorization, we see that Err and CoMI to have unclear trends in pCR, sometimes giving a weak signal for unlearning efficacy. Fgt and Err in pCR tests with sample sizes ranging from a tenth to half of the class give a substantial signal, but much weaker than IC test. Thus, targeted removal contributes to better measurements of forgetting for relatively smaller deletion sets, but has limited usefulness without adding confusion.

E.5. Varying Confused Classes

Throughout our experiments, we only confused the hardest pair of classes in the dataset (Cat and Dog for CIFAR10, Maple Tree and Oak Tree for CIFAR100). Here we ablate the chosen class pair, grouping the ten classes in CIFAR10 into five pairs to maximize diversity. The five pairs are arranged in increasing order of similarity below:

- Frog (6) - Horse (7)
- Bird (2) - Ship (8)
- Airplane (0) - Deer (4)
Table 13. Varying $|D_f|$ for IC test. Results are reported as Memorization|Property Generalization. In CIFAR10, at 10% of the class size, the IC test reliably detects imperfect forgetting across metrics. In CIFAR100, imperfect removal of memorization is detected at 10% of the class size, a noticeable effect on generalization requires a larger deletion set.

| Method   | Fgt (↓) | Err (↑↓) | CoMI (↑) |
|----------|---------|----------|----------|
| | 400   | 1K | 2K | 4K | 400 | 1K | 2K | 4K | 400 | 1K | 2K | 4K |
| Original | 348/211 | 854/300 | 1721/513 | 3016/927 | 88.8/19.8 | 88.3/24.7 | 85.0/34.2 | 76.6/53.9 | 91.4 | 86.2 | 80.6 | 66.1 |
| 10-layers CF | 247/181 | 543/335 | 1003/306 | 1226/335 | 63.2/18.4 | 54.8/22.6 | 50.6/25.1 | 30.9/26.8 | 83.9 | 76.9 | 71.8 | 58.7 |
| EU       | 209/181 | 487/233 | 920/296 | 1165/352 | 55.8/18.3 | 49.9/22.4 | 48.9/25.2 | 28.9/27.0 | 78.8 | 75.6 | 70.5 | 59.1 |
| 50-layers CF | 69/158 | 187/186 | 302/181 | 569/229 | 24.5/16.6 | 22.4/18.2 | 21.4/19.1 | 20.1/23.2 | 64.8 | 59.9 | 58.5 | 54.2 |
| EU       | 55/179 | 124/175 | 216/182 | 453/195 | 17.5/17.0 | 18.8/17.4 | 18.5/19.4 | 23.0/23.0 | 53.3 | 55.0 | 53.2 | 53.2 |
| Retrain  | 41/164 | 92/183 | 176/185 | 390/184 | 17.0/16.4 | 17.3/19.1 | 19.1/20.4 | 22.2/21.4 | 50.2 | 52.3 | 52.2 | 52.0 |

Table 14. Varying $|D_f|$ for CR test. Results are reported as Memorization|Property Generalization. The CR test is not able to reliably distinguish varying levels of property generalization and provides a weak signal for memorization, particularly for small $|D_f|$.

| Method   | Fgt (↓) | Err (↑↓) | CoMI (↑) |
|----------|---------|----------|----------|
| | 400   | 1K | 2K | 4K | 400 | 1K | 2K | 4K | 400 | 1K | 2K | 4K |
| Original | 40/37  | 99/39 | 196/46 | 395/70 | 100/0/37.0 | 99/0/40.0 | 97/0/43.0 | 100/0/57.0 | 88.2 | 88.5 | 85.6 | 86.4 |
| 10-layers CF | 35/32 | 81/41 | 152/35 | 325/57 | 90.0/35.5 | 81.0/41.5 | 78.0/38.5 | 85.8/51.0 | 86.5 | 83.8 | 78.9 | 79.0 |
| EU       | 28/33  | 61/33 | 124/40 | 290/60 | 80.0/39.0 | 72.0/38.0 | 68.0/41.5 | 76.8/53.0 | 86.4 | 78.9 | 76.0 | 72.1 |
| 50-layers CF | 5/36 | 19/37 | 37/28 | 86/36 | 20.0/40.0 | 37.0/36.5 | 37.0/38.5 | 41.8/44.5 | 55.1 | 64.4 | 56.3 | 59.0 |
| EU       | 3/32   | 11/34 | 31/32 | 49/37 | 32.5/36.5 | 34.0/41.0 | 38.5/37.0 | 40.5/44.5 | <50 | 52.4 | 55.4 | 52.0 |
| Retrain  | 3/30   | 10/30 | 29/24 | 64/31 | 35.0/35.5 | 39.0/36.0 | 31.5/32.5 | 39.0/39.5 | <50 | 52.7 | <50 | 53.0 |

- Automobile (1) - Truck (9)
- Cat (3) - Dog (5)

In table 15 we can see that the number of confused samples by any model is much higher as we go from left to right, indicating that confusing a similar pair of classes makes unlearning more difficult. Both memorization (left of ↓) and property generalization (right of ↓) trends across varying levels of unlearning, from Original to Retrain, are consistently preserved. This shows that irrespective of the chosen class pair, the IC test is able to clearly distinguish varying degrees of forgetting.
Table 15. Varying confused class pairs on CIFAR10, with the similarity of the classes increasing from left to right. Scores reported as Memorization|Property Generalization. While the IC test reliably detects imperfect forgetting across class pairs, the trends are clearer for more similar classes.

| Class Pair | {6,7} | {2,8} | {0,4} | {1,9} | {3,5} |
|-------------|-------|-------|-------|-------|-------|
| CIFAR10 \((D_f = 400)\) |       |       |       |       |       |
| Original    | 238/38 | 304/40 | 320/43 | 317/100 | 348/211 |
| 10-layers CF | 15/4  | 38/6  | 85/11 | 127/64 | 247/181 |
| EU          | 14/4  | 21/4  | 63/12 | 111/66 | 209/181 |
| 50-layers CF | 3/2   | 9/6   | 8/10  | 33/57  | 69/158  |
| EU          | 0/5   | 2/7   | 5/11  | 18/65  | 55/179  |
| Retrain     | 1/1   | 1/9   | 1/7   | 9/54   | 41/164  |
| CIFAR10 \((D_f = 4000)\) |       |       |       |       |       |
| Original    | 2980/1005 | 2855/868 | 2952/983 | 2915/1000 | 3016/927 |
| 10-layers CF | 57/7  | 76/14 | 83/18 | 305/93 | 1226/335 |
| EU          | 50/10 | 49/13 | 76/21 | 257/99 | 1165/352 |
| 50-layers CF | 24/6  | 30/7  | 31/11 | 160/70 | 569/229  |
| EU          | 12/4  | 21/6  | 27/12 | 130/66 | 453/195  |
| Retrain     | 3/1   | 19/6  | 20/17 | 117/65 | 390/184  |

Table 16. We compare using Warm Restarts and keeping a single learning rate cycle between the same maxLR and minLR. CoMI represents memorization while Fgt measures property generalization.

| Sched   | CoMI | Fgt |
|---------|------|-----|
| CIFAR10 \((D_f = 4000)\) |     |     |
| WR      | 58.66 | 335 |
| No      | 58.10 | 340 |
| CIFAR100 \((D_f = 400)\) |     |     |
| WR      | 77.99 | 58  |
| No      | 78.87 | 55  |

E.6. Learning Without Restarts

One concern which may arise is whether catastrophic forgetting performs well due to warm restarts in our learning rate schedule. We ablate this effect in Table 16 and see that in all cases removing warm restarts has no effect on the degree of catastrophic forgetting.