From Dormant to Deleted: Tamper-Resistant Unlearning Through Weight-Space Regularization

Shoaib Ahmed Siddiqui *
University of Cambridge

Adrian Weller
University of Cambridge
The Alan Turing Institute

David Krueger
Mila

Gintare Karolina Dziugaite
Google DeepMind
Mila

Michael C. Mozer
Google DeepMind

Eleni Triantafillou
Google DeepMind

Abstract

Recent unlearning methods for LLMs are vulnerable to relearning attacks: knowledge believed-to-be-unlearned re-emerges by fine-tuning on a small set of (even seemingly-unrelated) examples. We study this phenomenon in a controlled setting for example-level unlearning in vision classifiers. We make the surprising discovery that forget-set accuracy can recover from around 50% post-unlearning to nearly 100% with fine-tuning on just the *retain* set—i.e., zero examples of the forget set. We observe this effect across a wide variety of unlearning methods, whereas for a model retrained from scratch excluding the forget set (gold standard), the accuracy remains at 50%. We observe that resistance to relearning attacks can be predicted by weight-space properties, specifically, L_2-distance and linear mode connectivity between the original and the unlearned model. Leveraging this insight, we propose a new class of methods that achieve state-of-the-art resistance to relearning attacks[2].

1 Introduction

Machine unlearning is the problem of removing the influence of specific training datapoints, the *forget set*, from a pretrained model. This was initially motivated from the perspective of privacy, the right-to-be-forgotten [3, 33], data protection policies [32], and recently applied to a range of problems, including removing harmful knowledge [28, 46, 8].

Exact unlearning refers to completely eliminating the influence of the forget set. This objective can be achieved by retraining the model without those datapoints (i.e., *retrain from scratch*) [2]. However, such a method is computationally prohibitive as it requires retraining the model for every unlearning request [2]. This issue motivated the development of approximate unlearn-

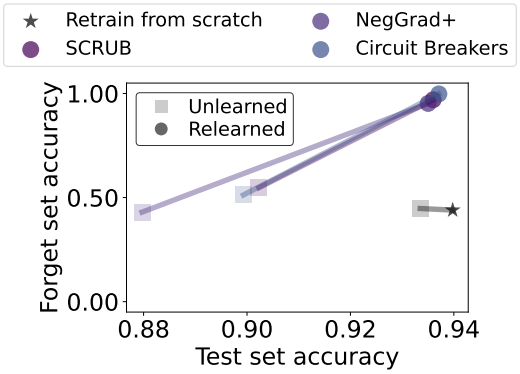

Figure 1: **Fine-tuning an unlearned model on just the retain set recovers performance on the forget set!** Results on CIFAR-10 using a forget set of atypical examples from class 'airplane'.

*Correspondence to msas3@cam.ac.uk.

[2]Code to reproduce our experiments: https://github.com/shoaibahmed/vision_relearning

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

ing methods, where the aim instead is to only approximately remove the influence of the given datapoints [5, 11, 26, 28, 46], in exchange for greater efficiency. While such methods may seemingly succeed at matching the retrained-from-scratch model on some simple metrics, like the accuracy on the forget set, it is unclear if they permanently remove the influence of these datapoints [16]. In fact, evaluating whether they do is a research problem in its own right [16, 42].

In this work, we take a different perspective, focusing on relearning attacks on unlearned models. We consider *tampering attacks* [4, 40], where the attacker is able to fine-tune the model weights. This direction is inspired by a growing set of observations in the context of unlearning "knowledge" in LLMs: fine-tuning an unlearned model even on seemingly benign data may cause the believed-to-be-unlearned knowledge to re-emerge [20, 31, 6, 4]. However, these studies are carried out under conditions that make it difficult to draw clear conclusions. First, these works study unlearning *knowledge, capabilities, or topics*, where the problem is inherently under specified. For example, given a dataset containing knowledge necessary for making bioweapons, the goal may be to fully remove the capability of constructing bioweapons, while preserving general knowledge of biology. In this setting, it is hard to draw a clear line between forbidden and permissible knowledge and pinpoint all training examples responsible for acquiring different types of knowledge. Furthermore, knowledge is hard to measure, especially due to nuances of natural language, requiring question-answer evaluation which may be sensitive to the particular phrasing. Finally, because LLMs can make complex inferences beyond the training set, not all knowledge that is extractable should be attributed to a failure of unlearning [38]. In many cases, the acquisition of knowledge is natural and unavoidable, hence, making it difficult to distinguish between the two in these problem settings.

To address these issues, we study relearning attacks in a setting that allows for controlled experimentation: unlearning specific training examples from (small) vision classification models, a problem for which a plethora of approximate unlearning methods have been developed and tested [14, 15, 13, 26, 11, 41, 45, 36]. In example-level unlearning, the gold standard is clear, namely, to retrain the model without the forget set. As our models are small enough, we can compute the gold-standard solution and use it as the reference point for comparison. Because we consider classification, we can also use accuracy as a simple and well-understood measure of performance. These properties combined allow us to compare the tamper resistance of different unlearning algorithms with the correct reference point, and therefore draw conclusions about their quality.

We evaluate a range of increasingly-complex unlearning algorithms in this setting and discover a surprising finding: for numerous unlearning algorithms, the accuracy of the forget set jumps from around 50% post-unlearning to nearly 100% after fine-tuning the unlearned models on *only the retain set*, which is disjoint from the forget set. Fig. 1 shows this phenomenon on CIFAR-10 using ResNet-18, after having attempted to unlearn a subset of atypical instances of class 'airplane'. We note that a model retrained from scratch without the forget set does not exhibit this behaviour, with the accuracy remaining at 50%. Therefore, the recovery of forget set accuracy can be safely interpreted as a failure of these algorithms to fully remove the influence of the datapoints in the forget set.

Our extensive analyses leads to multiple insights. First, unlearning and relearning of typical examples is trivial, and unlearning methods behave similarly to the gold standard. However, we see a stark contrast in their respective patterns of behaviour on a forget set of atypical examples. Furthermore, taking a weight-space view [12], we discover a key characteristic of unlearning algorithms that are better at resisting these attacks: they yield an unlearned model that is distant from the pretrained model in the weight-space. Based on this insight, we propose a new class of unlearning algorithms that are superior in terms of resisting relearning attacks by incorporating terms in their objective that encourage the unlearned model to move far away from the pretrained model in the weight-space.

To summarize, we make the following contributions in this work:

- We show that unlearning algorithms fail to *delete* the influence of the forget set, which stays *dormant* and can resurface by fine-tuning *even on just the retain set*.
- We identify a key characteristic of methods that are more robust against relearning attacks, namely: the unlearned model is distant from the pretrained model in weight space.
- Leveraging this insight, we propose a new class of unlearning methods that attempt to push the unlearned model far away from the pretrained model. These methods are significantly more robust against relearning attacks in comparison to unlearning methods that operate only at the output-level [26] or the representation-level [28, 46].

2 Background and Related Work

Unlearning. The problem of machine unlearning was introduced by [3]. The goal is to remove the influence of a "forget set" from a model that was trained on a dataset including that set. This was motivated by privacy and right-to-be-forgotten policies [32]. The perfect unlearning method, from the perspective of fully erasing the influence of the forget set, is to simply retrain the model excluding that set. However, the computationally prohibitive training costs make such an approach infeasible in most practical cases. [2] propose to shard the dataset, and train an ensemble model over it, allowing to selectively retrain only the affected parameters. However, the computational cost is still prohibitive in the worst case, while also leading to poorer performance in some cases due to the use of specialized architectures. These issues motivated the development of approximate methods that accept imperfect unlearning in exchange for greater efficiency. This is the family of methods we focus on in this work. The goal in approximate unlearning is to post-process the trained model as efficiently as possible in order to closely match the model which is retrained from scratch using only a small amount of model fine-tuning [21, 5, 11, 26, 28, 46, 45, 42]. This is a challenging problem as imperfect attempts to erase the influence of the forget set post-hoc may have a number of unwanted side-effects, such as harming the overall utility of the model [42].

Unlearning quality metrics. Since most approximate unlearning methods that are applicable to deep models do not come with theoretical guarantees about the quality of their approximation, we are required to estimate how well they approximate retraining from scratch empirically. This is a research problem in and of itself, and current rigorous metrics are very computationally expensive [42, 16]. Furthermore, unlearning entails fundamental trade-offs, such as between forgetting and maintaining the model's utility. This requires a multifaceted evaluation metric that captures relevant factors aside from forgetting quality. Commonly, in vision classification, the accuracy on the retain set and the accuracy on the test set are used to measure model utility. In similar spirit to our work, *time to relearn* [14] quantifies the strength of unlearning by the number of optimization steps required to reacquire forgotten information by directly fine-tuning on it. We instead show that we can restore forget set accuracy even when fine-tuning on only a subset of it, or solely on the retain set.

Re-emergence of attempted-to-be-unlearned knowledge via fine-tuning. Recent work in language models showed that believed-to-be-unlearned knowledge can re-emerge by fine-tuning on a small subset of the forget set or even on seemingly-unrelated data [20, 31, 6, 4]. Relatedly, it has also been shown that fine-tuning a language model on benign inputs can reverse the safety tuning of the model [34, 27]. A key distinction sets our work apart from all prior efforts. They study unlearning *knowledge, or capabilities*, rather than *specific training examples*. Their goal is to remove unwanted knowledge beyond the specific instances in the forget set, e.g., fully remove a dangerous capability (such as bioweapon construction) after having unlearned on a specific dataset containing related knowledge [28]. This problem is inherently less well-specified compared to unlearning specific examples where we have a clear definition of the ideal solution, namely, retraining from scratch without the specific examples. In LLMs, measuring knowledge is also nuanced, requiring question-answering tools, for instance, where the success of extracting knowledge may depend on the phrasing [47]. We study relearning attacks for example-level unlearning in vision classifiers, a setting where the forget set is well-specified and the goal is well-defined and simple to measure.

3 Problem Formulation

Let $\mathcal{D}_{tr}$ denote a training set and $\mathcal{A}$ a learning algorithm. Let $\mathcal{M}_P = \mathcal{A}(\mathcal{M}_I, \mathcal{D}_{tr})$ denote the "pretrained model", obtained by training on $\mathcal{D}_{tr}$, starting from a random initialization $\mathcal{M}_I$. Now, let $\mathcal{D}_F \subset \mathcal{D}_{tr}$ denote a forget set that we want to unlearn, and let $\mathcal{D}_R = \mathcal{D}_{tr} \setminus \mathcal{D}_F$ denote the retain set.

The goal of an unlearning algorithm $\mathcal{U}$, is to post-process the pretrained model $\mathcal{M}_P$ to remove the influence of $\mathcal{D}_F$. Specifically, we denote an unlearning algorithm by $\mathcal{U} : \mathcal{M} \times \mathcal{D}_R \times \mathcal{D}_F \mapsto \mathcal{M}$ that takes in a model, retain set $\mathcal{D}_R$, forget set $\mathcal{D}_F$, and returns an unlearned model $\mathcal{M}_U = \mathcal{U}(\mathcal{M}_P, \mathcal{D}_R, \mathcal{D}_F)$. Ideally, the unlearned model $\mathcal{M}_U$ should match the gold-standard "retrained-from-scratch" model $\mathcal{M}_{RS} = \mathcal{A}(\mathcal{M}_I, \mathcal{D}_R)$ which starts from a random initialization and trains on only the retain set, fully eliminating the influence of $\mathcal{D}_F$. We desire unlearning algorithms that can approximate that solution but are much more efficient than retraining from scratch.

In this work, we study relearning attacks that apply a further fine-tuning phase attempting to reintroduce the forget set. Such attacks that are able to modify the model's weights are referred to in the literature as *tampering attacks* [4, 40]. We carry out these attacks by fine-tuning the model on the union of $\mathcal{D}_R$ and a subset of "relearning examples" $\mathcal{D}_{F_{re}} \subset \mathcal{D}_F$. We denote the relearned model as $\mathcal{M}_{RL} = \mathcal{A}'(\mathcal{M}, \mathcal{D}_R \cup \mathcal{D}_{F_{re}})$, where $\mathcal{A}'$ denotes a fine-tuning algorithm used for relearning (which might be similar to $\mathcal{A}$ with slightly different hyperparameters) and $\mathcal{M}$ can be either $\mathcal{M}_U$ or $\mathcal{M}_{RS}$.

We measure performance on the held-out portion of the forget set (held-out from the perspective that it was not used during relearning), denoted $\mathcal{D}_{F_{ho}} = \mathcal{D}_F \setminus \mathcal{D}_{F_{re}}$. We vary the size of $\mathcal{D}_{F_{re}}$ and measure the effect on relearning. We also measure performance on a test set $\mathcal{D}_{te}$, to measure utility.

An ideal unlearning algorithm is one that is *tamper resistant*: upon relearning, its accuracy on the forget set does not increase more than it would by learning the relearning set anew. In other words, the forget set accuracy of $\mathcal{A}'(\mathcal{M}_U, \mathcal{D}_R \cup \mathcal{D}_{F_{re}})$ should not be higher than that of $\mathcal{A}'(\mathcal{M}_{RS}, \mathcal{D}_R \cup \mathcal{D}_{F_{re}})$. At the same time, an ideal unlearning algorithm would not sacrifice the test accuracy.

Threat model. Similar to tampering attacks considered for LLMs [40, 31, 20, 6, 4], we assume that the defender has access to a pretrained model, and performs unlearning using any algorithm of their choice. Furthermore, we assume that the attacker has white-box access to the unlearned model provided by the defender, the retain set $\mathcal{D}_R$, and limited access to the forget set (i.e., the relearning set $\mathcal{D}_{F_{re}}$). The goal of the attacker is to recover performance on the full forget set $\mathcal{D}_F$, while minimizing the number of unlearned examples needed $\mathcal{D}_{F_{re}}$ (as relearning becomes trivial if $\mathcal{D}_{F_{re}} = \mathcal{D}_F$). We also consider an extreme—and perhaps more realistic—case of access to only $\mathcal{D}_R$.

4 Experimental Setup

Models and Datasets. We use two different models for our evaluation from the ResNet model family [17], namely ResNet-18 and ResNet-34 [17]. In terms of datasets, we use CIFAR-10 and CIFAR-100 datasets [25] with 10 and 100 classes respectively, and a total of $50k$ training instances in each case ($5k$ instances per class for CIFAR-10, and 500 instances per class for CIFAR-100).

Evaluation. All models are evaluated in terms of accuracy on the held-out part of the forget set $\mathcal{D}_{F_{ho}}$ (same subset between all models), as well as the test set $\mathcal{D}_{te}$. While we can report accuracy on the full forget set $\mathcal{D}_F$ for the unlearned model, we instead report accuracy on the remaining subset in order to enable a direct comparison of the impact of the relearning attack. The line plots show accuracy every 10 optimization steps. When reporting results using a scatter plot, we average the test set accuracy as well as the accuracy on the forget set for the last 50 steps reported in the line plots.

Pretraining. We pretrain the model for 300 epochs using Adam optimizer [24] with a learning rate of 1e-4, cosine learning rate decay with a decay factor of 0.1, batch size of 128, and a weight decay of 1e-4 in all configurations.

Unlearning. We consider two unlearning settings: *sub-class unlearning*, where the forget set consists of 10% of the class instances (here, sub-class means a subset of the complete class), and *class-agnostic unlearning*, where we select 1% of the data set regardless of class labels. This ensures that we use the same number of examples in the forget set for both settings on CIFAR-10 (we only evaluate sub-class unlearning on CIFAR-100). We use a smaller learning rate of 1e-5 without any weight decay and optimize the model for a 100 epochs during the unlearning phase.

Relearning attack. During this phase, we fine-tune on a combination of the retain set $\mathcal{D}_R$ and a subset of the forget set for relearning ($\mathcal{D}_{F_{re}}$). We explore the impact of different choices for relearning examples in Appendix G. We again use a small learning rate of 1e-5 without any weight decay, and optimize the model for just 10 epochs (except Fig. 1 where we optimized the model for 300 epochs). Similar to the pretraining stage, we use a cosine learning rate decay with a decay factor of 0.1.

4.1 Baseline Unlearning Methods

We consider a range of different baseline unlearning methods. Each method has its own set of hyperparameters. We attempted to select the hyperparameters to achieve a good trade-off between the test accuracy and the forget set accuracy for each unlearning method.

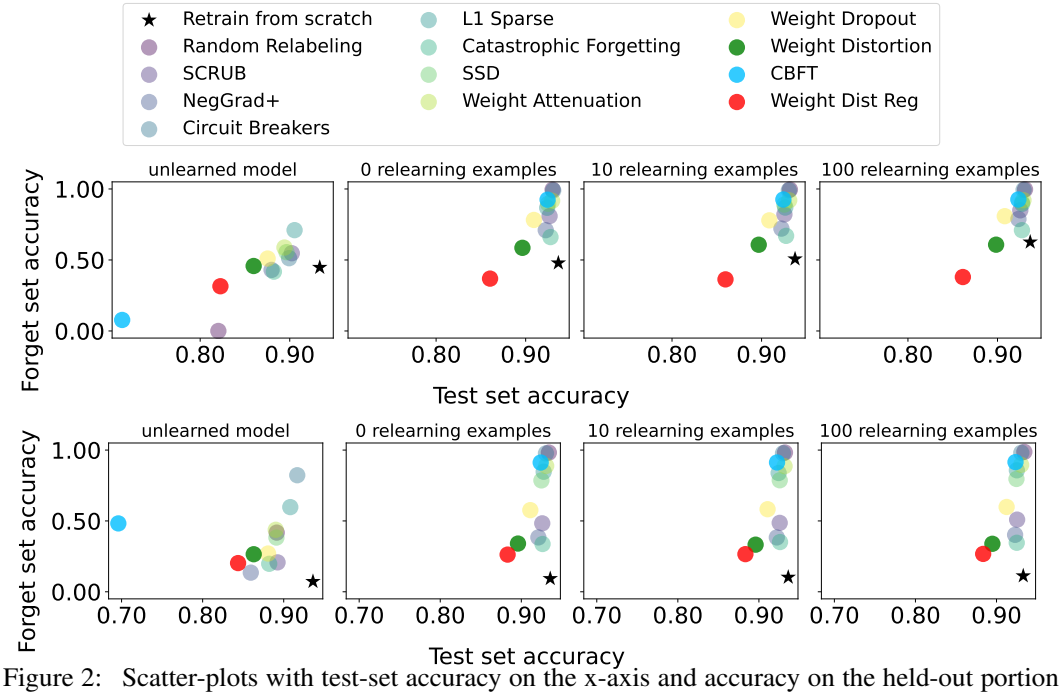

Figure 2: Scatter-plots with test-set accuracy on the x-axis and accuracy on the held-out portion of the forget set, $\mathcal{D}_{F_{ho}}$, on the y-axis. The left-most subplot indicates performance immediately following unlearning. The next three subplots are following a relearning attack with instances of the retain set, $\mathcal{D}_R$ and a varying number of instances of the forget set, $\mathcal{D}_{F_{re}}$ (0, 10, and 100, respectively). Each point is the average performance of the last 50 steps (see Fig. 11 for the whole trajectory for sub-class unlearning and Fig. 12 for the trajectory for class-agnostic unlearning). The forget set is comprised of atypical examples (from the 'airplane' class, i.e., sub-class unlearning for the top row and all classes, i.e., class-agnostic unlearning in the bottom row) in CIFAR-10. The figure indicates that many methods achieve near-perfect recovery of unlearned knowledge with only a small amount of model fine-tuning, even with 0 relearning examples (fine-tuning on only the retain set). Weight Distortion, CBFT, and Weight Dist Reg are introduced in Section 6.

SCRUB [26] uses a two-phase training procedure where it interleaves iterations on the forget set and the retain set. The loss function minimizes KL-divergence between the pretrained model and the unlearned model output distributions on the retain set, along with cross-entropy on the true labels. For unlearning, it maximizes the KL-divergence on the forget set between the distributions of the pretrained model and the unlearned model.

Circuit Breakers [46, 28] was proposed particularly to unlearn knowledge in language models. The training procedure attempts to push the representations apart by minimizing cosine similarity with the pretrained model on the forget set, while minimizing the Euclidean distance of the representations on the retain set to avoid model collapse. We apply circuit breaker loss on layer 4 and layer 7 of our models, which is motivated by the fractional depth considered in the original work.

NegGrad+ [26] maximizes the cross-entropy loss on the forget set, while minimizing the loss on the retain set. We used the alternating variant (similar to SCRUB) instead of joint optimization of the two losses, as it resulted in better test accuracy as well as lower forget set accuracy.

Catastrophic Forgetting [42] uses repeated fine-tuning on the retain set with a weight decay, which naturally leads to a decay in the magnitude of the parameters that are unimportant for the forget set. We use a weight decay of 0.001 for all our models.

L1-Sparse [21] is similar to the Catastrophic forgetting in our case, except that it minimizes L_1-norm instead of the L_2-norm employed in weight decay.

Selective Synaptic Dampening (SSD) [11] identifies model weights to dampen based on their importance for the retain set and forget set, quantified using the Fisher information matrix. In contrast to the original paper, we follow this process with fine-tuning on the retain set to be consistent with our other baselines, giving SSD a better shot at repairing test accuracy.

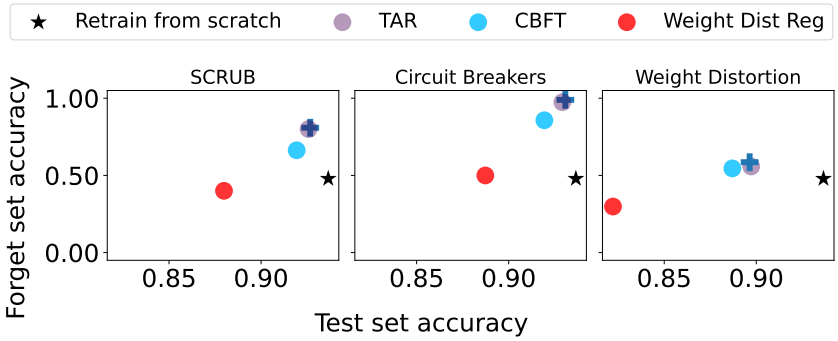

Figure 3: Comparison between test set accuracy and accuracy on the held-out part of the forget set $\mathcal{D}_{F_{ho}}$ after relearning, for subclass unlearning of atypical examples in CIFAR-10. We consider two-phase unlearning methods: first, an initial safeguard (unlearning phase) is applied, with the unlearning algorithm mentioned as the subplot title. Then, each of TAR, CBFT, and Weight Dist Reg are applied as a second phase for increasing the tamper-resistance. The '+' symbol represents the performance of the initial safeguard for reference. We observe that TAR fails to add any tamper-resistance in addition to that of the initial safeguard despite being designed for this.

Random Relabeling [15] relabels every example in the forget set with a random label. Note that we used an aggressive version of random relabeling where we re-assign a new label at every fine-tuning step. Hence, this can be considered analogous to minimizing divergence to a uniform distribution.

Weight Attenuation attenuates all model weights with a fixed attention factor of 0.5, followed by simple fine-tuning on the retain set.

Weight Dropout performs random (unstructured) pruning with a dropout factor of 0.2 (i.e., zeroing out 20% of the model weights), followed by simple fine-tuning on the retain set.

Tampering Attack Resistance (TAR) [40], originally proposed for LLM unlearning, defines a bi-level optimization, starting from an already unlearned model, with the aim to make it resistant against tampering attacks. It uses a first-order approximation of inner adversaries, and attempts to maximize the entropy of the model predictions after fine-tuning of the unlearned model. TAR also uses a representation alignment loss which minimizes the Euclidean distance between the representations of the initially unlearned model and the current model to avoid model collapse. Similar to Circuit Breakers, we apply the representation alignment loss on layer 4 and layer 7 of our models. Note that TAR relies on an unlearned model as a starting point, making it a two-phase approach.

5 Recent Unlearning Algorithms are not Tamper-Resistant

We begin by investigating the tamper-resistance of current state-of-the-art unlearning methods. We use CIFAR-10 and CIFAR-100 with ResNet-18, and attempt to unlearn instances of a single class ('airplane' class for CIFAR-10, and 'apple' class for CIFAR-100), or across classes (*class agnostic*).

The role of typicality for learning and unlearning. Not all examples are equally easy to unlearn, and not all examples are equally easy to (re)learn. Typical examples can be particularly easy to unlearn as even a no-op unlearning already yields similar predictions on these examples as the retrained from scratch model [21]. This is because they are, by definition, easy to predict regardless of whether they are part of the training set or not [10, 22, 1, 39]. Generally, because typical examples are more likely to be predicted correctly, they are easier to relearn as well. Based on this, we hypothesize that the typicality of examples is a key property that will determine relearning patterns and differences between unlearning algorithms compared to retraining from scratch.

To investigate this, we study three different settings, characterized by different typicality levels, where the forget set contains instances of the 'airplane' class that are: (i) most likely to be typical, (ii) randomly selected, and (iii) most likely to be atypical. We leverage pre-computed consistency scores from [22] to separate typical and atypical instances, by treating instances with the highest consistency scores as typical instances, while treating instances with the lowest consistency scores as atypical instances. Other scoring schemes are also equally applicable [10, 39, 1].

When the forget set consists of typical examples, relearning is trivial and uninteresting. The reason is that accuracy of the retrain-from-scratch model is essentially perfect on the forget set, even before relearning is applied, and all unlearning methods exhibit similar behavior (for the sake of completeness, we show the results for this case in Fig. 7 – Appendix A). Similarly, when the forget set consists of randomly selected examples, performance of retrain-from-scratch is nearly perfect because randomly selected examples are predominantly typical [10, 9, 1, 39] (again, for completeness, this case is shown in Fig. 8 – Appendix B). Consequently, we focus on atypical forget set items for the remainder of the paper; in this case, retrain-from-scratch will not predict correctly prior to relearning and thus we can measure the effectiveness of a relearning attack.

Relearning attacks succeed against several unlearning baselines. We present the results for sub-class unlearning for the forget set of atypical examples in Fig. 2 (top). In this and subsequent figures, existing methods are indicated by pastel colors. New methods, to be introduced shortly, are represented by saturated colors. For the most part, the existing methods all behave similarly and the reader need not attend to the individual methods. The accuracy of *retrain from scratch* is less than 50% on that forget set, and remains almost exactly the same when subjected to the relearning attack of fine-tuning only on the retain set (i.e., 0 relearning examples). The accuracy of this model shifts up slightly as we increase the number of relearning examples from the forget set (going from left to right). In stark contrast, the different unlearning methods we evaluate show a qualitatively different trend. We make the striking observation that some methods (such as Circuit Breakers, SCRUB, and Random Relabeling) are very susceptible to relearning attacks. For these methods, forget-set accuracy drops down after unlearning, near the desired reference point of the retrained model. However, upon relearning even on just the retain set, the model achieves near-perfect accuracy on the forget set—a jump from near 50% post-unlearning to nearly 100% after relearning.

Sub-class vs class-agnostic unlearning. We further attempt to understand if such differences exist in a class-agnostic forget-set setting (with the same number of forget-set examples, i.e., 500), which is now comprised of atypical examples across classes. Fig. 2 (bottom) shows that class-agnostic unlearning better differentiates among methods compared to sub-class unlearning. We see a wider spread among methods, even in the first subplot (post-unlearning, pre-relearning), since class-agnostic unlearning is harder. We also observe that while for sub-class unlearning, the performance of retrain-from-scratch (black star) on $\mathcal{D}_{F_{ho}}$ shifts upwards as more relearning examples are used, this shift does not occur with class-agnostic unlearning. This result is expected since relearning on a larger number of atypical examples does not improve performance on a disjoint set of other atypical examples (by definition of atypicality [10]). There is again a stark contrast between the performance of retrain-from-scratch model on $\mathcal{D}_{F_{ho}}$, where post-relearning accuracy remains very low, and many unlearning algorithms, where post-relearning accuracy again reaches nearly 100%. Importantly, the relative ranking between methods in the sub-class and the class-agnostic case remains consistent.

Relearning attacks succeed even against methods designed for tamper-resistance. We further compare two-phase methods that assume an initial unlearning round, followed by a subsequent training round in order to reduce susceptibility to relearning attacks. This is inspired by the methodology of TAR [40], which is explicitly designed to increase the resistance of the unlearned model against fine-tuning-based relearning attacks. Despite TAR's success in the case of language models, Fig. 3 shows that it struggles to provide any resistance against relearning attacks in our case.

6 A Weight-Space View on Understanding and Improving Tamper-Resistance

In the previous section, we demonstrated the susceptibility of existing prominent unlearning methods to tampering (Fig. 2). However, it is unclear what makes a method vulnerable or robust to these relearning attacks. In this section, we shed light on this question through a weight-space view. Specifically, we hypothesize that the susceptibility of an unlearned model to relearning may be associated with failing to move 'far enough' from the pretrained model in weight-space. We explore this hypothesis from two perspectives: (i) by measuring distances in weight-space, and (ii) through Linear Mode Connectivity analysis [12]. We then use these tools to interpret the tamper-resistance profiles of previously-proposed unlearning algorithms (Fig. 2) that show better tamper-resistance (such as Catastrophic Forgetting and L1 Sparse) compared to the ones that exhibit worse tamper-resistance (such as Random Relabeling, Circuit Breakers, and SCRUB).

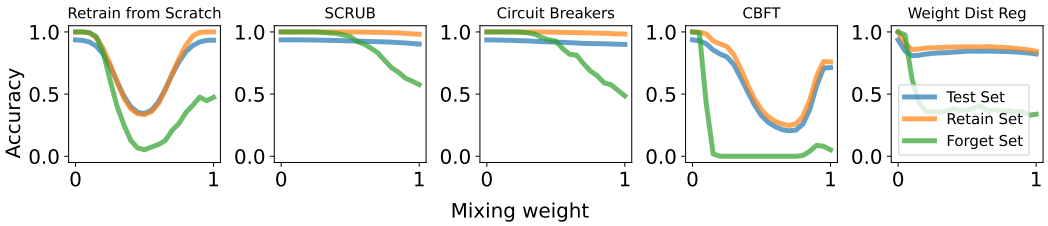

Figure 4: Linear mode connectivity analysis on CIFAR-10, where the forget set is comprised of atypical examples. We construct a linear path between the pretrained and the unlearned (or retrained-from-scratch) model by interpolating the model parameters and batch-norm statistics using different mixing weights (shown on the x-axis). We report accuracy on the y-axis. 0 on the x-axis represents the pretrained model, while 1 represents the unlearned or retrained model. Retrain-from-scratch is not linearly connected to the pretrained model, whereas for unlearning algorithms, the resulting unlearned model is in many cases still linearly connected to the pretrained one.

Weight-space distance. Fig. 5 plots L_2 distance between the pretrained model and the unlearned model. We see that existing methods that we previously showed to be susceptible to relearning (pastel colors) induce only small movement in parameter space, indicated by the small L_2 distance.

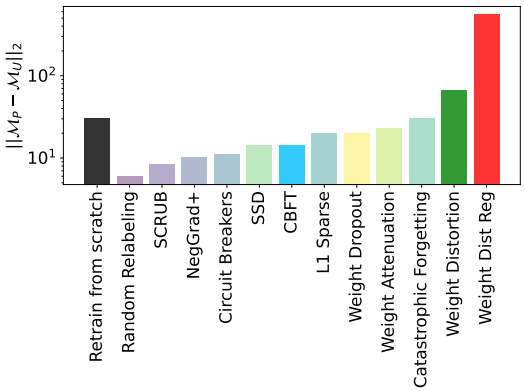

Figure 5: L_2 norm of the difference between the parameters of the pretrained and the unlearned models induced by different unlearning methods. We consider only the difference in the parameters, while ignoring the batch-norm statistics for ResNet-18 trained on CIFAR-10, where the forget set is comprised of atypical examples.

Generally, we see that methods with higher L_2 distance exhibit higher robustness against relearning attacks presented in Fig. 2. Notably, two methods based on these insights which we describe shortly, *Weight Distortion* and *Weight Dist Reg*, have the highest L_2-norm and higher tamper-resistance. We remark that, out of previous methods, those with increased tamper-resistance (such as Catastrophic Forgetting and L1 Sparse) have comparatively higher distance norm compared to methods with very poor tamper resistance (such as Random Relabeling, Circuit Breakers, and SCRUB). We note that both Catastrophic Forgetting and L1 Sparse use weight-space regularizers (L_2 and L_1, respectively). Fine-tuning without any regularizer failed to unlearn the forget set in our evaluations, which aligns with our weight-space interpretation. We consider more elaborate analysis regarding layers that are particularly important for robustness against relearning attacks as an interesting direction for future work.

Linear Mode Connectivity (LMC). We further investigate the relationship in weight-space between the unlearned and the retrained from scratch model via the lens of LMC [12]. We plot the accuracy when interpolating between the two models (where we interpolate both model parameters as well as model batch-norm statistics) along a linear path. We compare the pretrained and retrained from scratch model (which are trained from the same initialization), as well as the pretrained and different unlearned models in Fig. 4. Looking at the retrain from scratch plot, we see that there is a high-loss barrier between the two models, meaning that they are not in linearly connected modes. The same holds for methods with some tamper resistance (like Catastrophic Forgetting and L1 Sparse) but not for those vulnerable to tampering attacks like SCRUB, where we observe no such barrier.

6.1 A New Class of Tamper-Resistant Unlearning Methods

We now leverage these insights to propose a class of unlearning methods that are designed with tamper-resistance in mind. We achieve this through objectives that either aim to induce a large

distance in the weight-space, or a loss barrier between the pretrained and unlearned models. Hence, any method that directly or indirectly attempts to separate out the pretrained model and the unlearned model is an instantiation of this framework.

Weight Distortion. This is a very simple method that adds isotropic zero-mean Gaussian noise to all model weights with a fixed standard deviation of 0.02, followed by simple fine-tuning on the retain set. We hypothesize that the addition of noise facilitates moving away from the pretrained model.

Weight Dist Reg. We directly aim to maximize the distance between the pretrained and unlearned models, by explicitly adding a term that quantifies the Euclidean distance between the two models. We maximize this loss during training, while minimizing the loss on the retain set.

Connectivity-Based Fine-Tuning (CBFT). We employ the method from [30], which was originally proposed to obtain models focusing on distinct recognition mechanisms. We maximize the loss on the midpoint between the pretrained model and the current unlearned model on examples from the retain set as well as the forget set, while only minimizing the loss on the retain set for the unlearned model. This attempts to add a high-loss barrier in between, while still retaining model utility on the retain set for the final unlearned model. We use a small weighting factor of 0.001 on the loss maximization term. Similar to [30], we ignore the loss maximization term if the loss magnitude exceeds a value of 50.

Findings. Fig. 2 shows that Weight Distortion and Weight Dist Reg are significantly more tamper-resistant in comparison to prior approaches. However, we observe that CBFT is less effective than Weight Dist Reg and Weight Distortion across the board. We hypothesize that this is due to acting on model outputs and only indirectly influencing the weight-space. Indeed, while CBFT does create a larger loss barrier than other methods (Fig. 4), the L_2 norm between the pretrained and unlearned parameters is relatively low (Fig. 5). Overall, out of the two weight-space diagnostic tools, we find that the L_2 norm of the difference in model parameters is more reliable in predicting tamper-resistance. Fig. 3 shows that both Weight Dist Reg and CBFT, when applied as a second phase on an initial safeguard, can substantially improve its tamper-resistance, unlike TAR. The only exception is if the initial safeguard is Weight Distortion, which already has sufficient tamper-resistance.

7 Analysis and Discussion

On the trade-off between tamper-resistance and test accuracy. As discussed previously, there are inherent trade-offs in unlearning, between forgetting the specified examples while maintaining utility (measured via test accuracy) [26, 6, 42]. Here, we discover a different trade-off between resisting relearning attacks, and test accuracy. Indeed, Fig. 2 shows that the methods that are best at defending against these attacks are the ones with the lowest test accuracy. Having surfaced this fundamental tension, we hope future work improves on the current Pareto frontier formed by our new methods.

Findings hold across datasets/architectures. The results presented in the paper are consistent across models i.e., ResNet-34 on CIFAR-10 as presented in Fig. 10 – Appendix E. Furthermore, these results are also consistent across different datasets, i.e., CIFAR-100, as highlighted in Fig. 9 – Appendix C. Finally, our findings are also consistent on a higher-resolution Imagenette dataset [19] as highlighted in Table 1 – Appendix D.

Efficiency of Unlearning Algorithms. We used a fix budget of 100 epochs for unlearning to avoid confounding from differences in optimization efficiency. In order to justify the use of approximate unlearning algorithms instead of exact unlearning (i.e., *retrain from scratch*) with a training budget comparable to pretraining, we evaluate the efficiency of unlearning algorithms in Table 3 – Appendix I. The results show that our methods are much more efficient at unlearning (and consequently resisting re-emergence of knowledge) even after a single epoch of unlearning compared to other approaches.

Difference between Quantization and Relearning Attacks. Language models have been shown to be susceptible to quantization attacks [44], where quantization after unlearning is sufficient to recover unlearned knowledge from the model. We evaluate susceptibility to quantization in Fig. 6. Interestingly, our vision models are not vulnerable to these quantization attacks: forget set accuracy stays at the same level as the full-precision unlearned model regardless of the number of bits until it gets close to 8 bits, at which point the model collapses. Thus, the spontaneous-recovery phenomenon we explore via relearning attacks is distinct from the recovery obtained via quantization. We hypothesize that the primary reason why quantization attacks succeed in language models is due

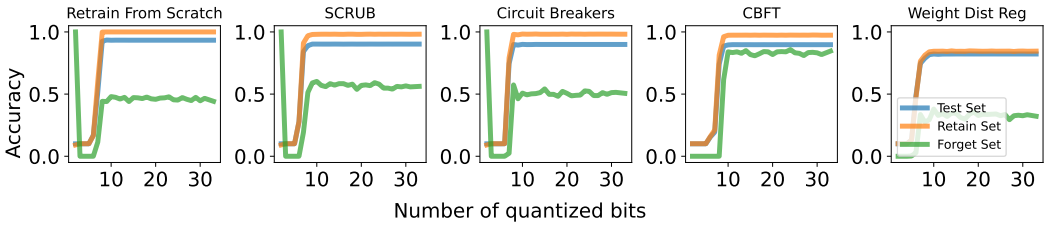

Figure 6: Comparison between the number of quantized bits and accuracy of the model for different unlearning methods on CIFAR-10 and ResNet-18, where the forget set is comprised of atypical examples. In contrast to language models, where quantization has been shown to be responsible for the re-emergence of unlearned knowledge [44], we observe that quantization has little to no impact on forget set accuracy until the point where full performance is recovered. Note that the accuracy of 100% at the start in most cases is an artifact of the model prediction collapsing to just class 0 (whose instances have been selected as our forget set).

to their larger number of parameters (on the order of 100 billion parameters [7, 29]), resulting in very small changes in weights during unlearning. Hence, models can easily revert to their previous values with quantization. This problem is further exacerbated with models that are trained with quantization in mind. Such models can immediately latch onto the previous values that were regularized to be easily quantizable [23]. Similar to our findings, [44] hypothesized that small differences are the cause of this susceptibility to quantization attacks. They further proposed increasing the learning rate as a potential mechanism to induce larger differences between the pretrained and unlearned models [44], while our work introduces more principled approaches for the same purpose.

The role of the retain set for relearning. Given the surprising finding that we can recover forget set accuracy while fine-tuning on only the retain set, we ask: are the retain set examples necessary for this to occur, or could we have used any other examples from the same distribution? We investigate this in Appendix G where we replace the retain set with different sets, and fine-tune the unlearned model on those different sets (combined with the 'relearning set', which is a subset of the forget set, as before). Notably, replacing the retain set with test examples (that the model was not previously trained on) causes the relearning effect to be a bit less pronounced, especially when 0, or few, relearning examples are used, highlighting the importance of using training rather than held-out data for inducing relearning. This observation relates to other recent findings on anticipatory knowledge reawakening when exposing the model to a repeated sequence of documents. In this scenario, as the model processes documents in a fixed order, it unexpectedly begins to recover an increasing amount of information about a previously seen example even before encountering that example again [43].

8 Conclusion

We showed that unlearning methods are susceptible to relearning attacks where the forget set accuracy can be recovered simply by fine-tuning on the retain set. For atypical examples in particular, there is a stark contrast between the relearning patterns of unlearning methods compared to retraining from scratch. Based on weight-space analysis, we suggest two diagnostic tools for understanding tamper-resistance, and propose simple methods that yield state-of-the art tamper-resistant results, revealing new pathways for improving unlearning. The authors of [36] argued that unlearning algorithms that operate on representations rather than at the level of outputs may be more robust at defending some types of attacks. Our findings take this discussion further, showing that *unlearning methods that operate at either the level of the model outputs or the representations, without any constraint on model weights* (which include methods such as SCRUB, Circuit Breakers, and Gradient Ascent, etc) *should struggle to induce robustness against relearning attacks.* On the other hand, methods that *directly or indirectly push the pretrained and the unlearned models apart via any intervention* (which includes distorting model weights, regularizing the model to decay parameter magnitude, or even directly pushing the models apart using an explicit loss term) *should be significantly more robust* to relearning attacks. Our proposed methods for increasing tamper-resistance exemplify this and we hope future work builds on these further.

Acknowledgements

The authors would like to acknowledge useful discussions with Yanzhi Chen and Ilia Shumailov regarding unlearning, and susceptibility to relearning. We are also very thankful to the anonymous reviewers for their useful feedback on the initial submission. AW acknowledges support from Turing AI Fellowship under grant EP/V025279/1, the Alan Turing Institute, and the Leverhulme Trust via CFI.

References

[1] Robert Baldock, Hartmut Maennel, and Behnam Neyshabur. Deep learning through the lens of example difficulty. *Advances in Neural Information Processing Systems*, 34:10876–10889, 2021.

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

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

[4] Zora Che, Stephen Casper, Robert Kirk, Anirudh Satheesh, Stewart Slocum, Lev E McKinney, Rohit Gandikota, Aidan Ewart, Domenic Rosati, Zichu Wu, et al. Model tampering attacks enable more rigorous evaluations of llm capabilities. *arXiv preprint arXiv:2502.05209*, 2025.

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

[6] Aghyad Deeb and Fabien Roger. Do unlearning methods remove information from language model weights? *arXiv preprint arXiv:2410.08827*, 2024.

[7] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407, 2024.

[8] Ronen Eldan and Mark Russinovich. Who's harry potter? approximate unlearning in llms. *arXiv preprint arXiv:2310.02238*, 2023.

[9] 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.

[10] 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.

[11] Jack Foster, Stefan Schoepf, and Alexandra Brintrup. Fast machine unlearning without retraining through selective synaptic dampening. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 12043–12051, 2024.

[12] Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning*, pages 3259–3269. PMLR, 2020.

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

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

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

[16] Jamie Hayes, Ilia Shumailov, Eleni Triantafillou, Amr Khalifa, and Nicolas Papernot. Inexact unlearning needs more careful evaluations to avoid a false sense of privacy. *arXiv preprint arXiv:2403.01218*, 2024.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[18] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

[19] Jeremy Howard. Imagenette. `https://github.com/fastai/imagenette`, 2019. Version 2. fast.ai.

[20] Shengyuan Hu, Yiwei Fu, Zhiwei Steven Wu, and Virginia Smith. Jogging the memory of unlearned model through targeted relearning attack. *arXiv preprint arXiv:2406.13356*, 2024.

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

[22] Ziheng Jiang, Chiyuan Zhang, Kunal Talwar, and Michael C Mozer. Characterizing structural regularities of labeled data in overparameterized models. *arXiv preprint arXiv:2002.03206*, 2020.

[23] Sangil Jung, Changyong Son, Seohyung Lee, Jinwoo Son, Jae-Joon Han, Youngjun Kwak, Sung Ju Hwang, and Changkyu Choi. Learning to quantize deep networks by optimizing quantization intervals with task loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4350–4359, 2019.

[24] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

[26] Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. Towards unbounded machine unlearning. *Advances in neural information processing systems*, 36, 2024.

[27] Simon Lermen, Charlie Rogers-Smith, and Jeffrey Ladish. Lora fine-tuning efficiently undoes safety training in llama 2-chat 70b. *arXiv preprint arXiv:2310.20624*, 2023.

[28] Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D Li, Ann-Kathrin Dombrowski, Shashwat Goel, Long Phan, et al. The wmdp benchmark: Measuring and reducing malicious use with unlearning. *arXiv preprint arXiv:2403.03218*, 2024.

[29] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

[30] Ekdeep Singh Lubana, Eric J Bigelow, Robert P Dick, David Krueger, and Hidenori Tanaka. Mechanistic mode connectivity. In *International Conference on Machine Learning*, pages 22965–23004. PMLR, 2023.

[31] Jakub Łucki, Boyi Wei, Yangsibo Huang, Peter Henderson, Florian Tramèr, and Javier Rando. An adversarial perspective on machine unlearning for ai safety. *arXiv preprint arXiv:2409.18025*, 2024.

[32] Alessandro Mantelero. The eu proposal for a general data protection regulation and the roots of the 'right to be forgotten'. *Computer Law & Security Review*, 29(3):229–235, 2013.

[33] Seth Neel, Aaron Roth, and Saeed Sharifi-Malvajerdi. Descent-to-delete: Gradient-based methods for machine unlearning. In *Algorithmic Learning Theory*, pages 931–962. PMLR, 2021.

[34] Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to! *arXiv preprint arXiv:2310.03693*, 2023.

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

[36] Nazanin Mohammadi Sepahvand, Eleni Triantafillou, Hugo Larochelle, Doina Precup, James J Clark, Daniel M Roy, and Gintare Karolina Dziugaite. Selective unlearning via representation erasure using domain adversarial training. In *The Thirteenth International Conference on Learning Representations*.

[37] 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.

[38] Ilia Shumailov, Jamie Hayes, Eleni Triantafillou, Guillermo Ortiz-Jimenez, Nicolas Papernot, Matthew Jagielski, Itay Yona, Heidi Howard, and Eugene Bagdasaryan. Ununlearning: Unlearning is not sufficient for content regulation in advanced generative ai. *arXiv preprint arXiv:2407.00106*, 2024.

[39] Shoaib Ahmed Siddiqui, Nitarshan Rajkumar, Tegan Maharaj, David Krueger, and Sara Hooker. Metadata archaeology: Unearthing data subsets by leveraging training dynamics. *arXiv preprint arXiv:2209.10015*, 2022.

[40] Rishub Tamirisa, Bhrugu Bharathi, Long Phan, Andy Zhou, Alice Gatti, Tarun Suresh, Maxwell Lin, Justin Wang, Rowan Wang, Ron Arel, et al. Tamper-resistant safeguards for open-weight llms, 2024. *URL https://arxiv. org/abs/2408.00761*.

[41] Reihaneh Torkzadehmahani, Reza Nasirigerdeh, Georgios Kaissis, Daniel Rueckert, Gintare Karolina Dziugaite, and Eleni Triantafillou. Improved localized machine unlearning through the lens of memorization. *arXiv preprint arXiv:2412.02432*, 2024.

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

[43] Yanlai Yang, Matt Jones, Michael C Mozer, and Mengye Ren. Reawakening knowledge: Anticipatory recovery from catastrophic interference via structured training. *arXiv preprint arXiv:2403.09613*, 2024.

[44] Zhiwei Zhang, Fali Wang, Xiaomin Li, Zongyu Wu, Xianfeng Tang, Hui Liu, Qi He, Wenpeng Yin, and Suhang Wang. Catastrophic failure of llm unlearning via quantization. *arXiv preprint arXiv:2410.16454*, 2024.

[45] Kairan Zhao, Meghdad Kurmanji, George-Octavian Bărbulescu, Eleni Triantafillou, and Peter Triantafillou. What makes unlearning hard and what to do about it. *Advances in Neural Information Processing Systems*, 37:12293–12333, 2024.

[46] Andy Zou, Long Phan, Justin Wang, Derek Duenas, Maxwell Lin, Maksym Andriushchenko, J Zico Kolter, Matt Fredrikson, and Dan Hendrycks. Improving alignment and robustness with circuit breakers. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[47] Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023.

A Tamper-Resistance with Typical Examples for the Forget Set

As highlighted in Section 5, typical examples can be particularly easy to unlearn as simply ignoring these examples during training (i.e., a no-op) yields similar predictions as the retrained from scratch model [21]. This is because they are, by definition, easy to predict regardless of whether they are part of the training set or not [10, 22, 1, 39]. Generally, because typical examples are more likely to be predicted correctly, they are easier to relearn as well. For the sake of completeness, we include the results for typical examples in Fig. 7. As evident from the figure, even the retrained from scratch model already achieves perfect accuracy on the forget set, highlighting that unlearning, and in turn, relearning are both trivial for typical examples. Therefore, we primarily focused on atypical examples in Section 5. We see that methods such as Random Relabeling, while completing forgetting the forget set, deviates in a very significant way from the retrain from scratch baseline, which is what we compare against.

B Tamper-Resistance with Random Set of Examples for the Forget Set

We further evaluate performance in the case of random selection of examples, rather than just typical examples as evaluated in Appendix A. We visualize these results in Fig. 8. Since randomly selected examples are predominantly typical [10, 9, 1, 39], retrain from scratch already achieves near-perfect accuracy. Furthermore, we observe only minor differences between methods. Consequently, we focused on atypical examples for the forget set in the main paper (Section 5) as these examples are hard to predict without being part of the training set, marking clearly the impact of relearning attacks which goes beyond model generalization.

C Tamper-Resistance on a More Complex Dataset (CIFAR-100)

We predominantly focus on CIFAR-10 dataset in the main paper (Section 5). We evaluate the susceptibility on a more complex CIFAR-100 dataset in Fig. 9 in order to understand the impact of the dataset. We only look at sub-class unlearning in this case, focusing on the 'apple' class for unlearning. Due to the higher complexity of the dataset, we observe lower test accuracies as evident from the scale of the x-axis. Retrain from scratch baseline achieves a low forget set accuracy. Furthermore, we see a wider spread of different methods on this more complex dataset, with the trend looking similar to the class-agnostic results presented in Fig. 2 (as it provided a more complex forget set comprising of the most atypical examples from the dataset). It is interesting to note that CBFT [30], which was not tamper-resistant on CIFAR-10, demonstrated significant resistance on CIFAR-100.

D Tamper-Resistance on a Higher-Resolution Dataset (Imagenette)

All our prior evaluations focused on small resolution images, i.e., 32×32 for both CIFAR-10 and CIFAR-100. In order to understand the generalizability of our approaches to higher resolution images, we evaluate on the higher-resolution ImageNette dataset [19], which is a subset of the ImageNet dataset [35]. The results are presented in Table 1. Similarly to our findings on both CIFAR-10 (Section 5) and CIFAR-100 (Appendix C), we see that the accuracy of *retrain from scratch* on the forget set after relearning stays close to 50%, while the accuracy of unlearning algorithms is significantly higher, except for our proposed *weight dist reg* algorithm among the tested methods.

E Tamper-Resistance of a Larger Model

All our prior results focused on the smaller model, i.e., ResNet-18 [17]. In order to understand the impact of model size, we evaluate the susceptibility to relearning attacks on ResNet-34 (with almost double the number of parameters compared to ResNet-18, going from $\sim 11M$ to $\sim 21M$). The results are visualized in Fig. 10. Note that we directly transfer the hyperparameter settings from our ResNet-18 experiments. Hence, we found that Weight Dist Reg, which was the most competitive method in terms of robustness, became susceptible to relearning. It is worth noting that the test set accuracy as well as the forget set accuracy for Weight Dist Reg in this case deviates significantly from all prior results, where it demonstrated consistently lower accuracies, highlighting a potential

Model	Label	Test Acc (%)	Forget Set Acc (%)
Unlearned	Retrain from Scratch	87.34%	67.53%
Unlearned	SCRUB	85.81%	85.71%
Unlearned	Weight Distortion	83.16%	74.03%
Unlearned	Weight Dist Reg	83.72%	84.42%
0 Relearn	Retrain from Scratch	86.40%	63.53%
0 Relearn	SCRUB	86.47%	93.40%
0 Relearn	Weight Distortion	84.71%	78.86%
0 Relearn	Weight Dist Reg	84.31%	73.87%
10 Relearn	Retrain from Scratch	86.73%	64.16%
10 Relearn	SCRUB	86.65%	94.21%
10 Relearn	Weight Distortion	84.62%	81.09%
10 Relearn	Weight Dist Reg	84.83%	75.77%

Table 1: Test and forget set accuracy across unlearning and relearning conditions when evaluating on the higher-resolution Imagenette dataset [19] with atypical examples for the forget set. These results are primarily consistent with our findings on both CIFAR-10 as well as CIFAR-100 datasets, where models learned with our proposed unlearning algorithm are much more robust against relearning attacks.

deficiency of the selected hyperparameters. Catastrophic Forgetting and Weight Dist Reg exhibit high robustness against relearning, which are instantiations of our framework.

F Evolution of Forget Set Accuracy During Relearning

In order to complement the results in Fig. 2 (top) and Fig. 3, we visualize the forget set accuracy evolution in Fig. 11. We see that methods that are most stable are nearly unaffected by increasing training time, highlighting that more training time might not be sufficient to increase their susceptibility further.

Similarly, Fig. 12 complements the results on class-agnostic unlearning in Fig. 2 (bottom), where we visualize both the line plots as well as the results from two-phase training strategies. We visually observe best separation between different methods on the line plot in this case of class-agnostic unlearning.

G Relearning with Examples from the Same Distribution, which are Distinct from the Retain Set

While we demonstrated that unlearned knowledge can be recovered through repeated fine-tuning on the retain set, it remains unclear whether we can also recover performance on the forget set by fine-tuning on new examples from the same data distribution that are unseen by the model. The question that we ask is: *are the same examples that the model was trained on particularly important for relearning, or other examples from the same data distribution equally effective?*

In order to simulate this scenario, we use examples from the test set (which are unseen by the model) for relearning instead of the retain set. We further evaluate using a corrupted version of the test set from CIFAR-10-C [18]. In particular, we use the JPEG corrupted examples with the highest severity level of 5. Note that we still use examples from the relearning set when considering scenarios where the number of relearning examples is > 0.

The results are presented in Fig. 13. We observe that using the retain set $\mathcal{D}_R$ achieves higher accuracy on the forget set compared to other choices. However, these differences diminish as the number of relearning examples increases, since these examples directly represent the unlearned knowledge and may become the dominant factor in recovery. When the test set is used for relearning (instead of the retain set), we unsurprisingly observe perfect accuracy on the test set, as it serves as the training set

Unlearning Method	MIA Acc (%)
Retrain from Scratch	52.2%
L1 Sparse	54.2%
SCRUB	51.6%
Circuit Breakers	55.0%
Gradient Ascent	54.2%
Random Relabeling	94.8%
Catastrophic Forgetting	54.2%
SSD	53.8%
Weight Attenuation	53.6%
Weight Distortion	53.2%
Weight Dropout	54.8%
Weight Dist Reg	48.4%

Table 2: Membership-Inference Attack (MIA) accuracy, where 50% indicates chance-level performance. Note that most of the methods are close to chance-level performance, except random relabeling. This indicates that MIA attacks are not sufficient to understand susceptibility to relearning attacks.

in this scenario. These results indicate that relearning is more effective when using examples that were seen during training, rather than a held-out set of examples.

These findings are also related to the other recent findings on anticipatory knowledge recovery when exposing the model to a repeated sequence of documents. In this scenario, as the model processes documents in a fixed order, it unexpectedly begins to recover an increasing amount of information about a previously seen example even before encountering that example again [43].

H Membership-Inference Attacks (MIA) on Unlearned Models

Membership-Inference Attacks (MIA) attempt to evaluate if it is possible to identify instances that were part of the model's training distribution from held-out instances [37]. Being able to distinguish instances leaks private information regarding an instance's membership. We follow the setup from [26] for membership-inference attacks with balanced loss-threshold-based classifier. The results are presented in Table 2. We observe that, as is the case in our initial findings with the forget set accuracy metric, this particular type of MIA can give a false sense that unlearning successfully erased all traces of unlearned data, even in cases where we know that the unlearned model is prone to relearning attacks. Note that we evaluated a particularly simple variant of MIA based on prior work [26]. Therefore, it is possible to use more complex variants of MIA, which might be better able to predict this vulnerability against relearning.

I Understanding the Efficiency of Unlearning Algorithms

In order to control for efficiency effects of the unlearning algorithm, all previous results were reported at a fixed unlearning budget of 100 epochs, which is comparable to the budget allocated for pretraining. This raises concerns regarding the utility of approximate unlearning algorithms, as one should use the gold-standard unlearning algorithm, i.e., *retrain from scratch* in cases where the computational budget is directly comparable to the pretraining budget.

In order to understand the significance of this selection, we vary the number of unlearning epochs and report test accuracy and forget set accuracy after relearning, while using the same set of hyperparameters. The results are presented in Table 3. It is clear from the table that while both catastrophic forgetting and weight distortion are successful in resisting relearning at the maximum budget of 100 epochs, catastrophic forgetting is still susceptible to relearning with a smaller number of unlearning epochs. On the other hand, weight distortion is more resistant to relearning even after a single epoch of unlearning.

Model	Method	Unlearning Epochs	Test Acc (%)	Forget Set Acc (%)
Unlearned	Catastrophic Forgetting	1	93.79%	100.00%
Unlearned	Catastrophic Forgetting	3	93.42%	100.00%
Unlearned	Catastrophic Forgetting	10	92.52%	94.50%
Unlearned	Catastrophic Forgetting	30	91.68%	77.75%
Unlearned	Catastrophic Forgetting	50	90.25%	60.00%
Unlearned	Catastrophic Forgetting	100	88.23%	41.75%
Unlearned	Weight Distortion	1	84.49%	43.50%
Unlearned	Weight Distortion	3	86.21%	49.00%
Unlearned	Weight Distortion	10	86.27%	52.75%
Unlearned	Weight Distortion	30	86.29%	53.25%
Unlearned	Weight Distortion	50	86.31%	50.50%
Unlearned	Weight Distortion	100	85.97%	45.75%
0 Relearn	Catastrophic Forgetting	1	93.71%	99.99%
0 Relearn	Catastrophic Forgetting	3	93.65%	99.99%
0 Relearn	Catastrophic Forgetting	10	93.37%	99.28%
0 Relearn	Catastrophic Forgetting	30	93.22%	93.63%
0 Relearn	Catastrophic Forgetting	50	93.08%	82.16%
0 Relearn	Catastrophic Forgetting	100	92.80%	66.14%
0 Relearn	Weight Distortion	1	90.14%	61.07%
0 Relearn	Weight Distortion	3	90.14%	61.66%
0 Relearn	Weight Distortion	10	89.86%	61.38%
0 Relearn	Weight Distortion	30	89.62%	60.69%
0 Relearn	Weight Distortion	50	89.44%	59.64%
0 Relearn	Weight Distortion	100	89.64%	58.59%

Table 3: Test and forget-set accuracy of the unlearned model after relearning attack on CIFAR-10 with atypical forget set examples. It is evident from the results that even a small number of unlearning epochs are sufficient to achieve robustness against relearning attack with *weight distortion*. In contrast, other methods such as *catastrophic forgetting* only work at the largest epoch setting. Note that all previous results were reported with 100 unlearning epochs, masking any impact of the efficiency of the unlearning algorithm.

J Computational Resources

We used NVIDIA RTX 3090 for each of our experiments, with the GPU equipped with 24GB of high-bandwidth memory (HBM) – we only use a tiny fraction of it as we train small ResNet models on CIFAR-10/100.

The pretraining takes about 3 hours. Each unlearning method takes about 2.5 hours, except TAR which takes about 9 hours. Each setting required training 21 unlearned models (3×3 combinations for two-phase training as we evaluate three methods with three different initial safeguards). This equates to about $18 \times 2.5 + 9 \times 3 = 72$ hours per setting. We evaluated 6 main settings: ResNet-18 on CIFAR-10 (typical, atypical, and random subset for the forget set), ResNet-34 on CIFAR-10 (atypical), ResNet-18 on CIFAR-10 with class-agnostic unlearning (atypical), and ResNet-18 on CIFAR-100 (atypical). Hence, this equates to about 450 hours. Grid evaluation of each model further takes one hour, resulting in an additional 130 hours (21×6 models).

Finally, taking into account all further evaluations as well as failed experiments, we expect to have invested about 1000 GPU hours on the project.

K Societal Impact

We highlight limitations of existing unlearning techniques in the context of visual recognition, showing that such techniques are prone to relearning attacks, where unlearned knowledge can be easily recovered with access to only retained knowledge. Deployment of such methods therefore

poses risks in the context of privacy or model safety. We further propose a new class of methods that are more robust against such attacks.

Developing better tamper-resistance to relearning attacks for unlearned models attempts to reduce the potential harm from these existing systems.

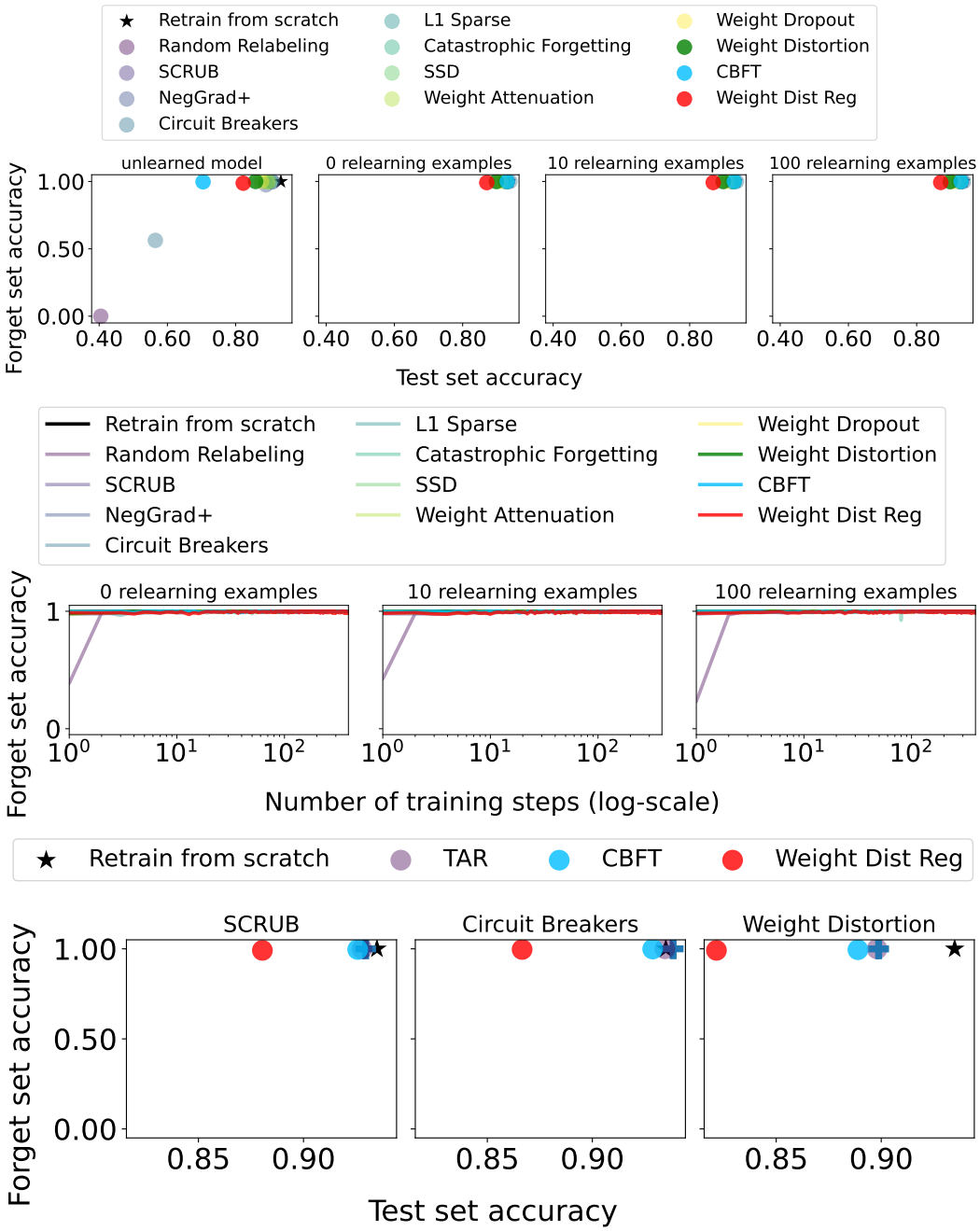

Figure 7: Comparison between test set accuracy and accuracy on the held-out part of the forget set $\mathcal{D}_{F_{ho}}$ on CIFAR-10 and ResNet-18, where the forget set is comprised of **typical examples** from the 'airplane' class. The figure indicates that all methods achieve perfect recovery of unlearned knowledge, which is consistent with the retrain from scratch baseline, as the model can directly generalize to these examples without having them as part of the training set by definition.

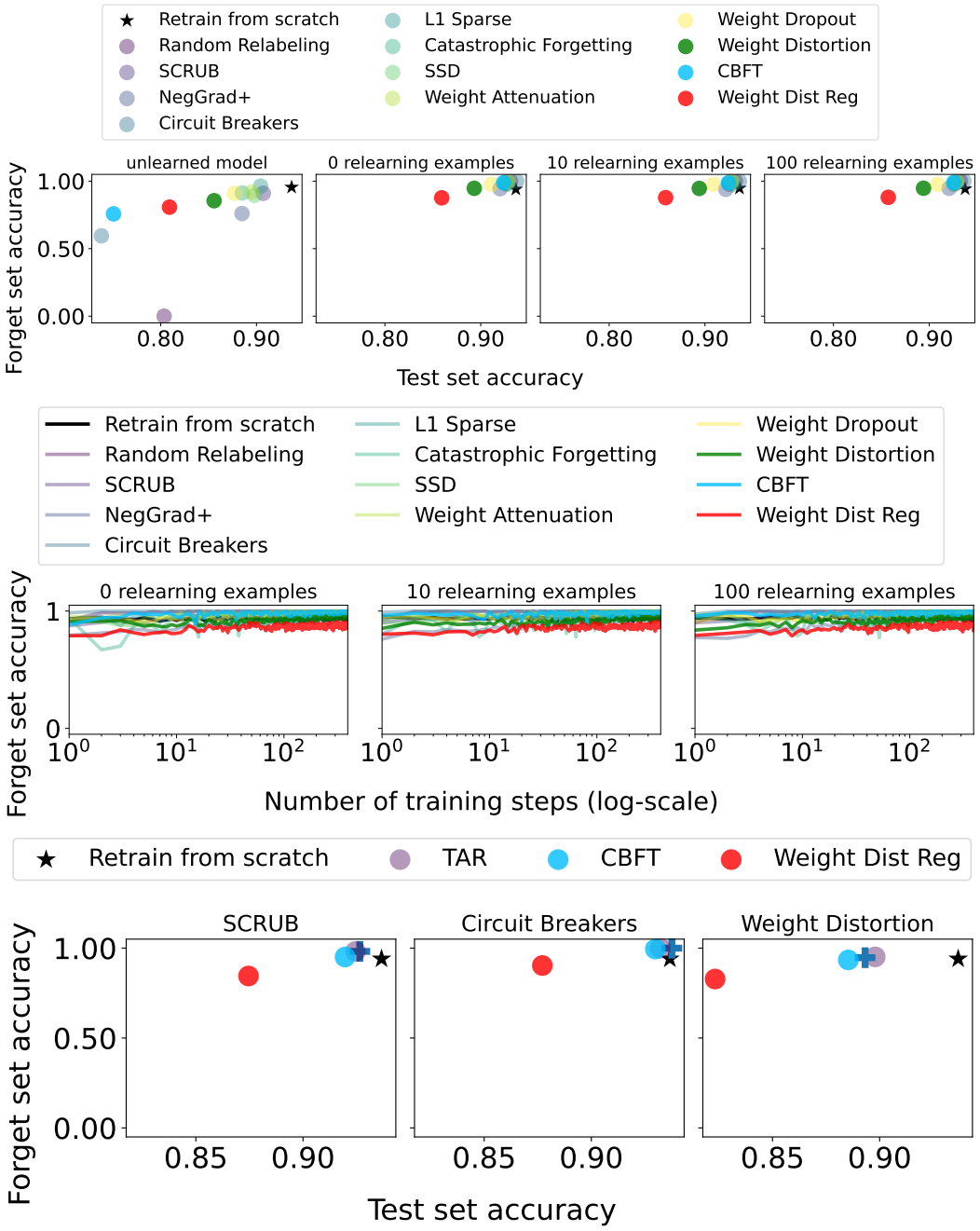

Figure 8: Comparison between test set accuracy and accuracy on the held-out part of the forget set $\mathcal{D}_{F_{ho}}$ on CIFAR-10 and ResNet-18, where the forget set is comprised of a **random subset** of the examples from the 'airplane' class. The figure indicates that retrain from scratch baseline can already achieve near-perfect accuracy, as the examples in a class are predominantly comprised of typical examples.

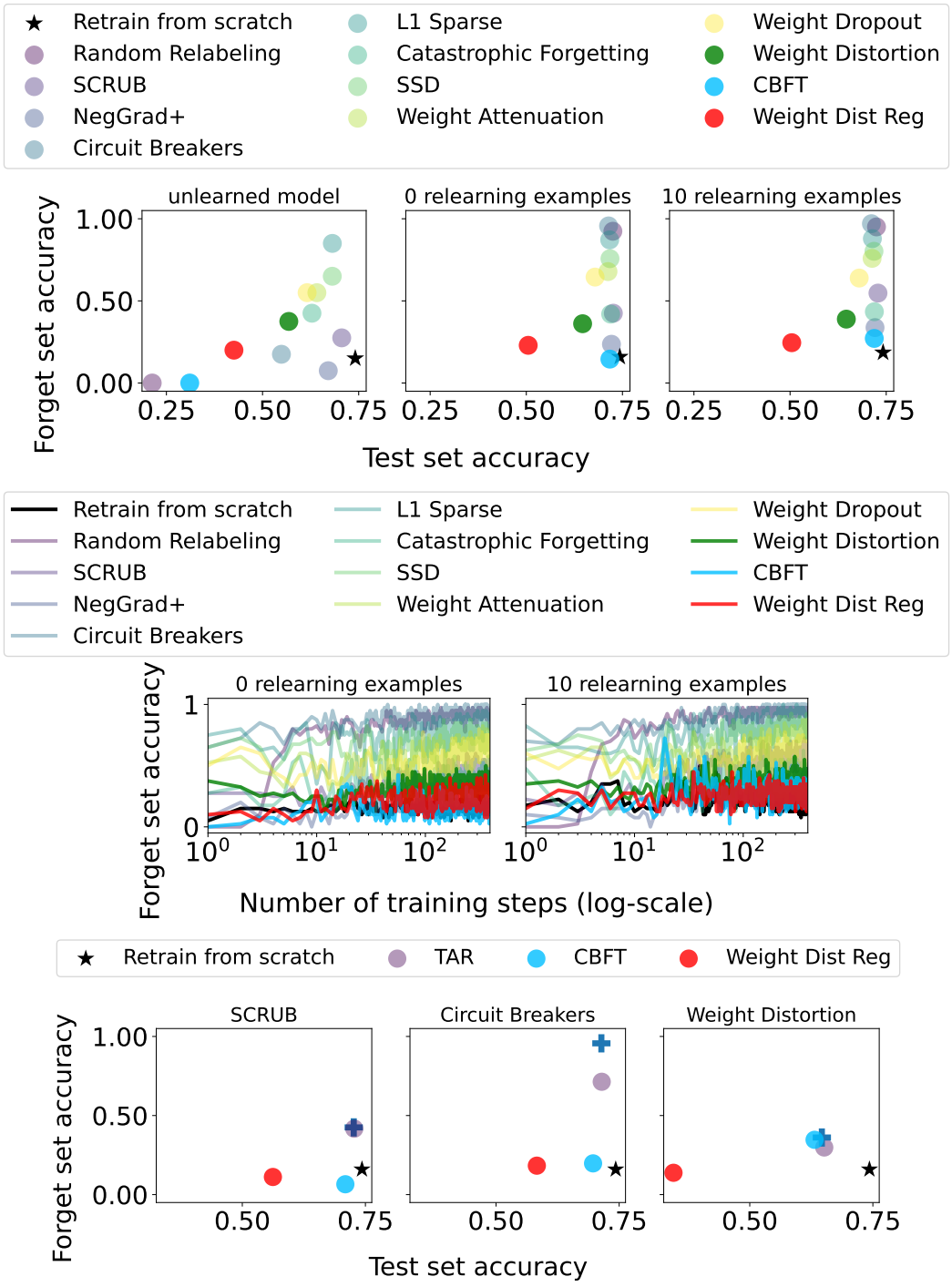

Figure 9: Comparison between test set accuracy and accuracy on the held-out part of the forget set $\mathcal{D}_{F_{ho}}$ on **CIFAR-100** and ResNet-18, where the forget set is comprised of atypical examples from the 'apple' class. We see significant variation among different methods on the more complex CIFAR-100 dataset, which is comparable to the results we observed with class-agnostic unlearning on CIFAR-10 in Fig. 2 (bottom).

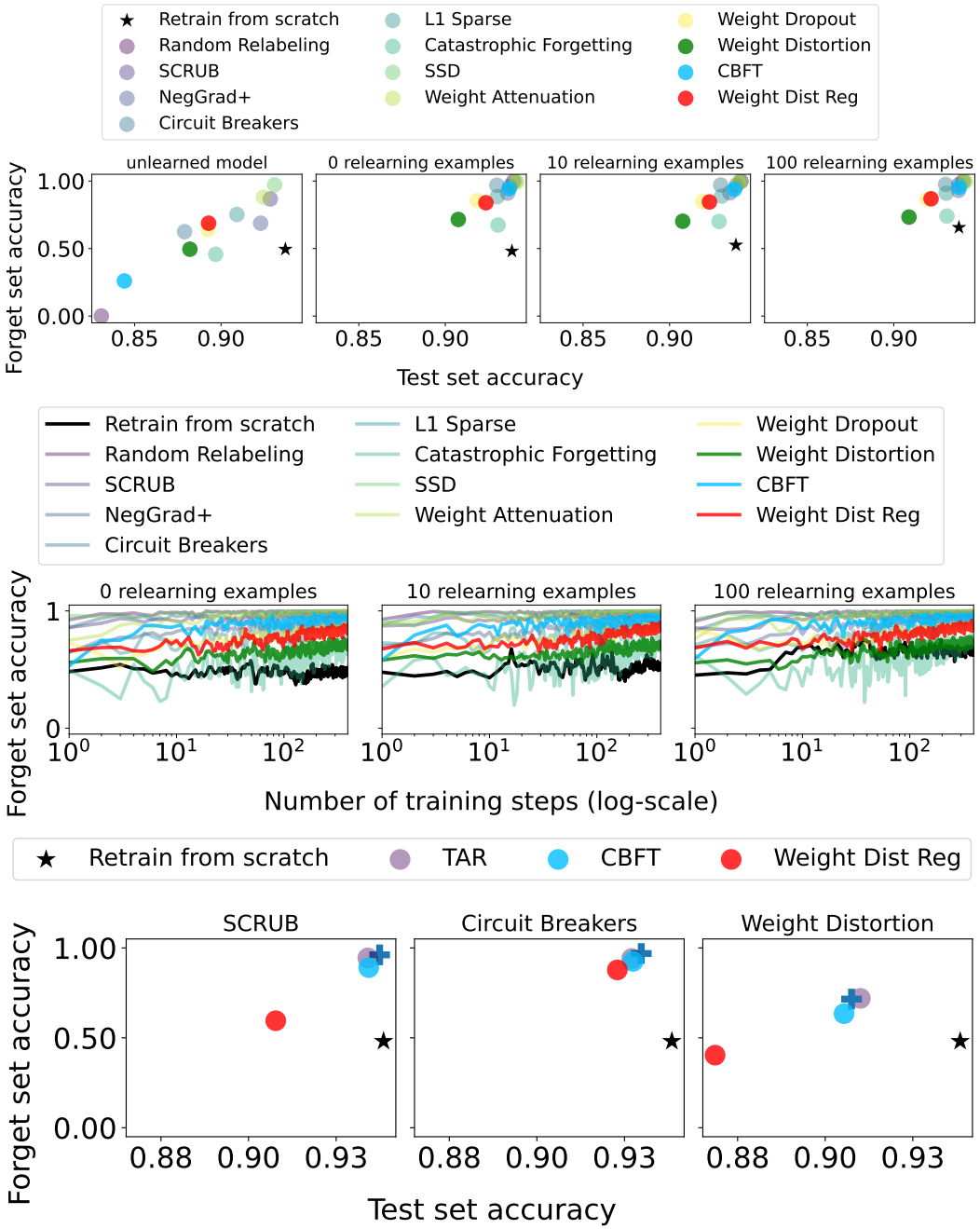

Figure 10: Comparison between test set accuracy and accuracy on the held-out part of the forget set $\mathcal{D}_{F_{ho}}$ on CIFAR-10 and **ResNet-34**, where the forget set is comprised of atypical examples from the 'airplane' class. We still see the distinction between robust and non-robust methods. However, the relative ranking between different methods changes as we use the same set of hyperparameters, which should be tuned considering a larger number of parameters in the model. We find that using a large model, i.e., ResNet-34 instead of ResNet-18 induces a slight shift in relative method ranking, particularly for methods that require careful tuning of hyperparameters.

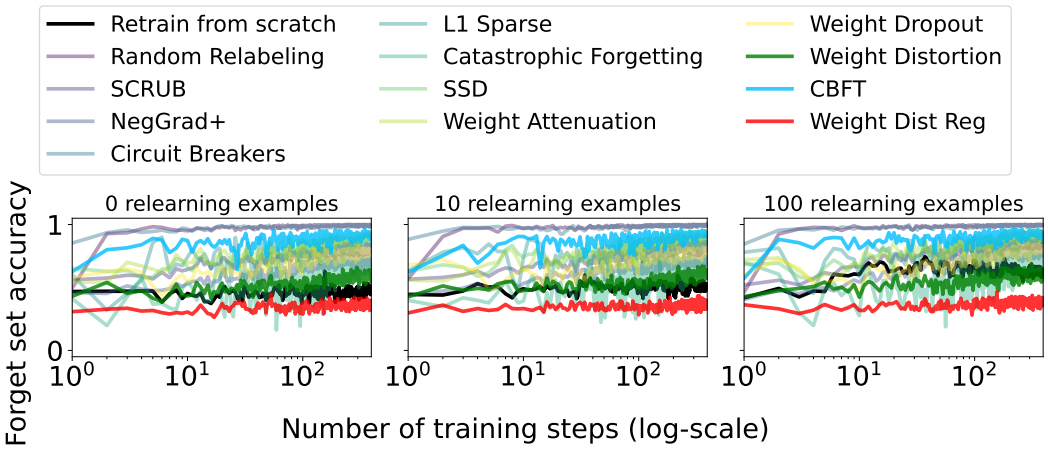

Figure 11: Comparison between relearning time and accuracy on the held-out part of the forget set $\mathcal{D}_{F_{ho}}$ on CIFAR-10 and ResNet-18, where the forget set is comprised of atypical examples from the 'airplane' class. The figure indicates that many methods achieve near-perfect recovery of unlearned knowledge with only a small amount of model fine-tuning, without even assuming access to the unlearned examples. This figure is an extension of the results presented in Fig. 2 (top).

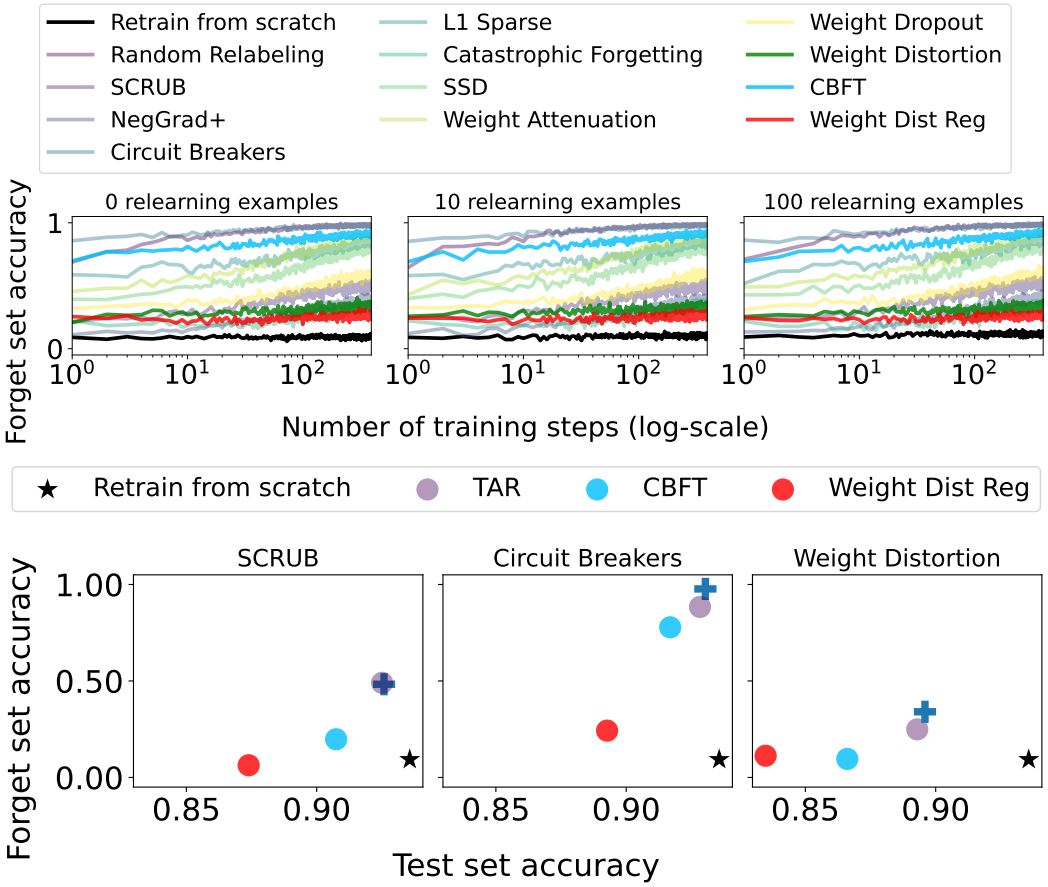

Figure 12: Comparison between test set accuracy and accuracy on the held-out part of the forget set $\mathcal{D}_{F_{ho}}$ on CIFAR-10 and ResNet-18, where the forget set is comprised of atypical examples from all classes. This figure is an extension of the results presented in Fig. 2 (bottom). We find that atypical examples, when selected from all the classes, are significantly harder in comparison to examples selected from a particular class, and hence, achieve a better separation between different methods in comparison to sub-class unlearning in Fig. 2 (top).

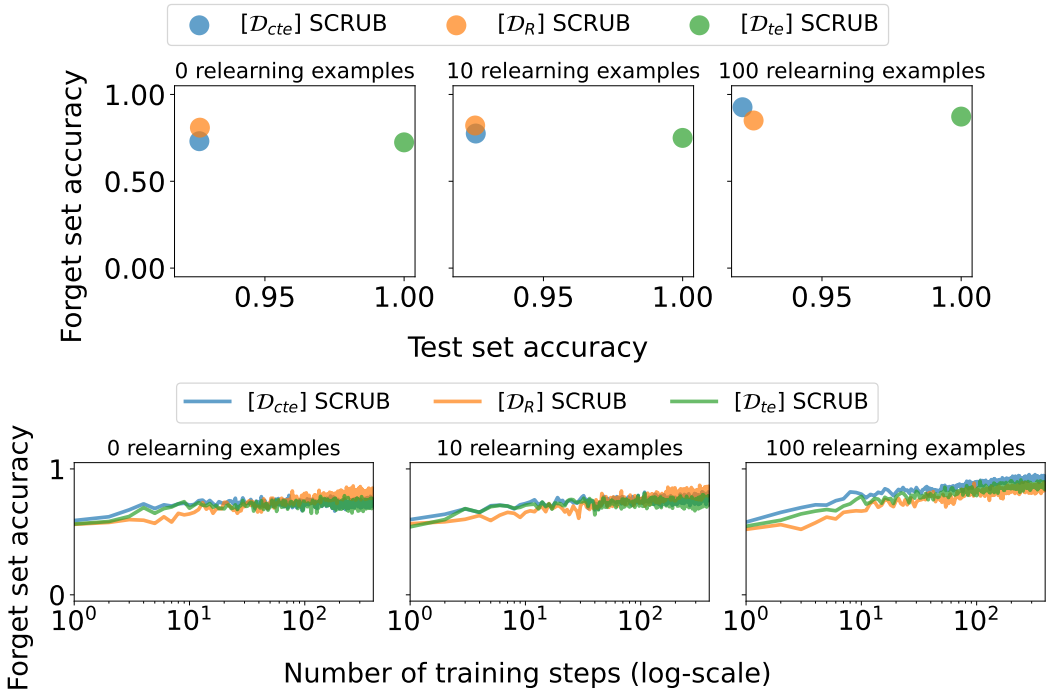

Figure 13: Comparison between training time and accuracy on the held-out part of the forget set $\mathcal{D}_{F_{ho}}$ on CIFAR-10 and ResNet-18, where the forget set is comprised of atypical examples, and using SCRUB as the unlearning method. We evaluate the use of different sets to remind the model, including the retain set $\mathcal{D}_R$, test set $\mathcal{D}_{te}$, corrupted test set $\mathcal{D}_{cte}$ taken from CIFAR-10-C [18] (using JPEG compression with a severity level of 5), along with the selected number of relearning examples. The figure indicates that there can be significant differences in relearned model accuracy based on the selected examples used for reminding the model.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: We highlight that the aim of the paper is to evaluate and propose new methods that are resistant against tampering attacks.

   Guidelines:

   - The answer NA 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 NA 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: [Yes]

   Justification: We highlight the limitations including the trade-off between clean accuracy and the tamper-resistance in Section 7. We also highlight the limitation of one instantiation of our framework (Mode Connectivity) in Section 6.

   Guidelines:

   - The answer NA 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: [NA]

Justification: The paper is based on empirical evidence, as we primarily evaluate the limitations of existing approximate unlearning methods.

Guidelines:

- The answer NA 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: We list all the main hyperparameters and settings in the paper, which are sufficient to reproduce our results.

   Guidelines:

   - The answer NA 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: [Yes]

Justification: We will provide code to reproduce our main results along with a README file. We use standard datasets i.e., CIFAR-10 and CIFAR-100 which should be directly accessible.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.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://nips.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, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We list all these minor details, including method-specific hyperparameters.

Guidelines:

- The answer NA 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: [No]

Justification: Since we are testing vulnerability, which is quite stable, we only did a single run. While extending it to multiple runs is desirable, this would incur significant additional cost without any major insight as the differences are already high.

Guidelines:

- The answer NA 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: [Yes]

   Justification: We list details about the hardware used for experiments and experimental runtime in Appendix J.

   Guidelines:

   - The answer NA 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: The conducted research does comply with NeurIPS code of Ethics.

   Guidelines:

   - The answer NA 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: [Yes]

    Justification: We highlight the societal impact of our work in Appendix K.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We don't release data or models that pose such risks.

    Guidelines:

    - The answer NA 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: We use publicly available datasets and model architectures.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: The paper does not release any new assets.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: No crowdsourcing and research with human subjects.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: No crowdsourcing and research with human subjects.

    Guidelines:

    - The answer NA 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 rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: LLM was not used for the core method development.

    Guidelines:

    - The answer NA 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 (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.