CORRECTIVE MACHINE UNLEARNING

Shashwat Goel[1]* **Ameya Prabhu**[2,3]* **Philip Torr**[2] **Ponnurangam Kumaraguru**[1] **Amartya Sanyal**[4]

[1]IIIT Hyderabad [2]University of Oxford [3]Tübingen AI Center, University of Tübingen
[4]Max Planck Institute for Intelligent Systems, Tübingen

ABSTRACT

Machine Learning models increasingly face data integrity challenges due to the use of large-scale training datasets drawn from the internet. We study what model developers can do if they detect that some data was manipulated or incorrect. Such manipulated data can cause adverse effects like vulnerability to backdoored samples, systematic biases, and in general, reduced accuracy on certain input domains. Often, all manipulated training samples are not known, and only a small, representative subset of the affected data is flagged.

We formalize "Corrective Machine Unlearning" as the problem of mitigating the impact of data affected by unknown manipulations on a trained model, possibly knowing only a subset of impacted samples. We demonstrate that the problem of corrective unlearning has significantly different requirements from traditional privacy-oriented unlearning. We find most existing unlearning methods, including the gold-standard retraining-from-scratch, require most of the manipulated data to be identified for effective corrective unlearning. However, one approach, SSD, achieves limited success in unlearning adverse effects with just a small portion of the manipulated samples, showing the tractability of this setting. We hope our work spurs research towards developing better methods for corrective unlearning, and offers practitioners a new strategy to handle data integrity challenges arising from web-scale training.We release our code at https://github.com/drimpossible/corrective-unlearning-bench.

1 INTRODUCTION

Foundation models are increasingly trained on large and diverse datasets, including millions of web pages and contributions from numerous users and organizations (Schuhmann et al., 2022; Gao et al., 2020). However, data integrity issues significantly impact model performance (Konstantinov & Lampert, 2022; Paleka & Sanyal, 2023) by introducing systemic biases (Prabhu & Birhane, 2021) and adversarial vulnerabilities (Barreno et al., 2006; Sanyal et al., 2021). For instance, a small manipulated subset of web data sources has led to large-scale model poisoning (Carlini et al., 2023), underscoring the vulnerability of these models to such adversarial tactics. Moreover, a critical real-world obstacle is that model developers can often only identify a fraction of the manipulated data, especially when the manipulations are small, imperceptible changes to input or incorrect labels.

Model developers may be notified of the manipulated data, either through poisoning defenses and other methods for monitoring of the data pipeline (Breck et al., 2019; Wang et al., 2019; Northcutt et al., 2021b) or external information. Due to high costs incurred in training, they may wish to update models trained on the corrupted data, instead of stopping their use. To solve this problem of removing the influence of manipulated data from a trained model, we introduce the concept of *Corrective Machine Unlearning*. This approach aims to efficiently eliminate any detrimental effects from the identified samples, even when the precise nature and extent of the manipulation is unknown. Corrective unlearning has different underlying requirements from the traditional unlearning literature (see Nguyen et al. (2022) for a survey) which is motivated by catering to user data deletion requests in light of privacy regulations (Council of European Union, 2018; California State Leglisature, 2018; Parliament of Canada, 2018). Specifically, corrective unlearning procedures do not need to obtain privacy guarantees on the "unlearned" data. Instead, they *must improve clean-label accuracy on parts*

*denotes equal contribution

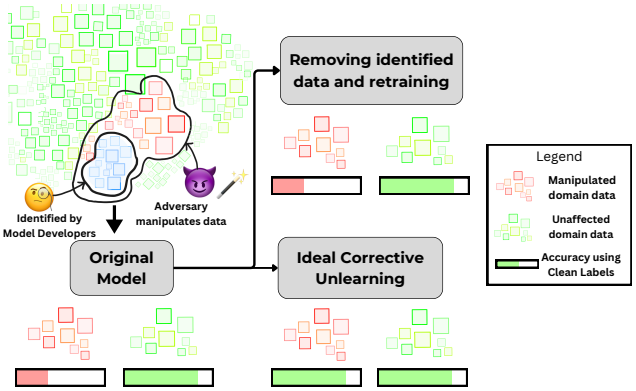

Figure 1: Traditionally, retraining after removing identified data is considered a gold standard in unlearning. However, since developers may not identify all the wrong data for unlearning, retraining-from-scratch on remaining data leads to poor clean-label accuracy. Ideally, corrective unlearning procedures should improve accuracy on the affected domain with access to only a representative subset of the wrong data.

of the data domain where model performance is adversely affected by the manipulated data while only having access to a representative subset of manipulated samples.

We investigate the application of state-of-the-art unlearning procedures (Kurmanji et al., 2023; Goel et al., 2023; Chundawat et al., 2023b; Foster et al., 2023) to remove adverse effects of two different kinds of manipulations. First, we study a classic poisoning attack (Gu et al., 2019), where a trigger pattern is embedded in a subset of samples, which are then assigned incorrect labels. Such manipulations occur when collecting both features and labels from internet web-pages which adversaries can modify, such as Wikipedia, as demonstrated by Carlini et al. (2023). This can lead to a backdoor where adversaries trigger model misclassifications by inserting the trigger pattern during deployment. Such actions can significantly harm applications, such as autonomous driving (Han et al., 2022). Second, we study the Interclass Confusion test (Goel et al., 2023) where the adversary incorrectly labels samples between two classes thereby entangling the model's representations. Such mislabeling can cause systematic biases in model outputs (Prabhu & Birhane, 2021). Such label-only manipulations can occur when model developers have their own unlabelled datasets but rely on external sources for annotation.

Model developers may eventually recognize compromised data sources and wish to unlearn the influence of this data from previously trained models. We find that many recent unlearning methods, including the traditional gold standard of retraining-from-scratch, fail in the context of corrective unlearning as illustrated in Figure 1. Particularly, even knowing 80% of the manipulated data is not enough to remove the adverse effects introduced by manipulating just 1% of the whole training data. However, the Selective Synaptic Dampening (Foster et al., 2023) method is able to remove the effect of BadNet poisoning with just 10% of the manipulated data being identified, showing the tractability of this setting. However, it leads to a significant drop in overall test accuracy, and fails in the Interclass Confusion setting, leaving much to be desired. Overall, our work highlights the need for unlearning procedures tailored to removing the influence of manipulated data.

2 Ideal Corrective Unlearning

In this section, we formalize the requirements of corrective unlearning, and detail key differences from the traditional privacy-oriented unlearning.

Problem Setting

We initiate our discussion by detailing the ideal corrective unlearning framework, introducing a precise threat model, and identifying specific desiderata.

Scenario: Training sets for large models are often compilations of data from diverse sources such as web pages, platforms like Reddit, data contractors, annotators, user inputs etc. These sources can introduce systematic biases or, more critically, contain data that has been adversarially manipulated,

motivating model developers to use corrective unlearning. Crucially, corrective unlearning methods should tackle a strong adversarial threat model that allows arbitrary manipulations. In doing so, it's reasonable to expect these unlearning methods can also address problems stemming from naturally occurring benign errors.

Threat Model: Next, we discuss the adversary and model developer's perspective.

Adversary's Perspective: The adversary can arbitrarily manipulate any portion of the input data, including labels in supervised learning scenarios. For example, in poisoning attacks, a trigger is inserted into each manipulated data sample, altering its label to an incorrect one (Han et al., 2022).

Developer's Perspective: Model developers identify some of the compromised data sources after having already trained a model, either through internal monitoring or defenses or external information like tipoffs. While detecting all manipulated data is challenging, it is feasible to be given a small subset which we assume to be representative of the broader set of manipulated data. Since the adversary can apply arbitrary manipulations, the exact manipulation type is unknown to the model developer apriori. The goal of model developers is to remove the adverse effects of the manipulated data from the original model using this small identified representative subset.

Formalization and Notation: Let $\mathcal{X}$ be the data domain, $\mathcal{Y}$ be the label space, and $\mathcal{P}$ be the distribution on $\mathcal{X} \times \mathcal{Y}$. Let $S_{tr} \subset \mathcal{X}$ be the training data, and $S_m \subset S_{tr}$ be the training samples manipulated by the adversary, either by modifying features, the associated training labels, or both. Let $\mathcal{D}_m \subset \mathcal{X}$ be the domain where performance is adversely affected when learning using S_m. For example, in poisoning, $\mathcal{D}_m$ contains samples with the poison trigger. In Interclass Confusion, $\mathcal{D}_m$ consists of samples from the two affected classes. Clearly, $\mathcal{D}_m$ also contains S_m. Finally, let A be the learning algorithm, and $M_o = A(S_{tr})$ be the original trained model.

A corrective unlearning algorithm U_{corr} "improves" the original model (M_o) by removing the influence of S_m. We expect only a subset of samples to be identified as manipulated, which we denote as the deletion set $S_f \subseteq S_m$. Thus, U_{corr} takes as inputs M_o, S_{tr}, S_f and yields an *unlearned* model M_u. Next, we list the goals of an unlearning procedure.

Desiderata: A corrective unlearning procedure U_{corr} has the following objectives:

(1) *Removing the influence of manipulated samples*: The primary goal is to remove the adverse effect learnt due to the manipulated data S_m. We operationalize this as improving the *clean-label accuracy on* $\mathcal{D}_m$:

$$\mathbb{E}_{(x,y)\sim\mathcal{P}}\left[\mathbb{I}\{h(x) = y\} \mid x \in \mathcal{D}_m\right]$$

where $h = U_{corr}(M_0, S_{tr}, S_f)$. We also compute the clean-label accuracy on the manipulated training set S_m to check if the unlearning procedure "corrects" the manipulation in the training data. It is important to note that while the domain $\mathcal{D}_m$ may be easier to identify for some kind of manipulations like poisoning, it may be more difficult in other cases.

(2) *Maintaining model utility*: Intuitively, the unlearning process should not harm performance on unrelated samples i.e. data outside $\mathcal{D}_m$, retaining model utility. We operationalize this as the overall accuracy ($\mathcal{X} \setminus \mathcal{D}_m$):

$$\mathbb{E}_{(x,y)\sim\mathcal{P}}\left[\mathbb{I}\{h(x) = y\} \mid x \notin \mathcal{D}_m\right]$$

where $h = U_{corr}(M_0, S_{tr}, S_f)$.

This quantity should decrease minimally, and can potentially increase due to a possibly conservative estimate of $\mathcal{D}_m$. For example, the manipulated data may affect the representations learned by the model in unintended ways and thereby impact the utility on unrelated and unexpected parts of the domain.

(3) *Effectiveness with Incomplete Identification*: Corrective unlearning algorithms (U_{corr}) should effectively unlearn adverse effects of manipulations even when the identified subset of the manipulated data S_f is a small representative subset of S_m. This means achieving (1), (2) even when $\frac{|S_f|}{|S_m|}$ is less than one.

(4) *Computation Efficiency*: The time taken by the procedure should be minimized.

We refer to the associated numbering explicitly throughout the rest of the paper.

DIFFERENCES FROM PRIVACY-ORIENTED UNLEARNING

Traditional unlearning seeks to ensure *retrain indistinguishability*: the unlearning procedure U aims to produce a distribution of models that is indistinguishable from one obtained without the forget set. Thus, for some learning algorithm A' which may be different from the original training procedure A, U should produce an indistinguishable distribution of models $U(M_o, S_{tr}, S_f) \sim A'(M_o, S_{tr} \setminus S_f)$. We highlight the distinctive aspects of corrective unlearning as opposed to traditional privacy-focused unlearning, and describe how these differences necessitate changes in unlearning evaluations and method design.

NO PRIVACY REQUIREMENTS

Key Distinction: In the corrective unlearning context, S_f and S_m does not need to be privatized, setting it apart from traditional unlearning.

Implications: Traditional unlearning is designed to meet strict privacy standards, necessitating either : (1) algorithms with theoretical privacy guarantees (Thudi et al., 2022) akin to those provided by differential privacy (Gupta et al., 2021), or at least (2) strong performance against privacy auditing on the data to be forgotten S_f (Golatkar et al., 2020a) such as those performed by Membership Inference Attacks (Shokri et al., 2017). Goel et al. (2023) argue rigorous empirical evaluations of the retrain indistinguishability goal are computationally infeasible for deep learning models. Not only is producing a distribution of models expensive, but since A' can differ from the original training procedure, there is a need to search the algorithm space for an A' that produces models indistinguishable from the unlearning procedure. Corrective unlearning bypasses these challenges by setting the practical goal of achieving empirical improvements in model accuracy on samples from the affected domain as the primary success metric (1).

REMOVAL OF INCORRECT TRAINING DATA

Key Distinction: The goal of traditional unlearning is to remove untampered but sensitive user data. However, corrective unlearning removes the influence of samples which were manipulated, either in data, labels or both. This can be particularly challenging for mislabeled data or in multi-class problems, where the corresponding clean version of the data and/or the correct label is unknown.

Implications: Removing accurate samples in traditional unlearning typically degrades model performance (Golatkar et al., 2020a). Moreover, some unlearning method explicitly try to randomize model outputs on forget set samples (Chundawat et al., 2023b; Li & Ghosh, 2023). However, in corrective unlearning, removing manipulated samples should significantly enhance model performance on parts of the affected domain $\mathcal{D}_m$ (1). It may also improve the quality of learned representations and thus increase overall accuracy (2).

RETRAIN-FROM-SCRATCH IS NO LONGER A GOLD STANDARD

Key Distinction: In traditional unlearning, all the data whose influence is to be removed from the model is specified by user deletion requests. However, when identifying manipulated data, it is unrealistic to assume all of it will be found. Thus, in corrective unlearning, $S_{tr} \setminus S_f$ will continue to have manipulated data from $S_m \setminus S_f$ (3).

Implications: Retraining from scratch on $S_{tr} \setminus S_f$ is the gold standard for traditional unlearning but it is computationally expensive. Therefore, the core challenge for traditional unlearning procedures is achieving computational efficiency (4). However, in corrective unlearning, as $S_{tr} \setminus S_f$ continues to have manipulated data, unlearning procedures that solely rely on it (Schelter, 2020; Bourtoule et al., 2021; He et al., 2021; Graves et al., 2021; Goel et al., 2023) perpetuate the adverse effects of the manipulation. This necessitates a methodological inquiry beyond computationally efficient approximations of *retraining from scratch*, which ceases to be a gold standard. This naturally leads to the question *How can we effectively remove the detrimental impacts of S_m using a representative, albeit smaller, subset S_f?*

Objective	Measurement	Poisoning Figure	IC Test Figure
Removing influence of manipulation on unseen samples (1)	Clean-label accuracy on test set samples from affected domain ($\mathcal{D}_m$)	Figure 2	Figure 3
Removing wrong predictions on manipulated training samples (1)	Clean-label accuracy on manipulated training samples (S_m)	Figure 4 (Appendix)	Figure 5 (Appendix)
Utility (2)	Accuracy on test set samples from unaffected domain ($\mathcal{X} \setminus \mathcal{D}_m$)	Figure 7 (Appendix)	Figure 6 (Appendix)

Table 1: Summary of figures as quantities reported on the Y-axis, with the X-axis varying $|S_f|$.

3 Experiments

We study image classification as the existing unlearning literature is situated here, only changing the task to corrective unlearning. We benchmark existing unlearning methods in the corrective unlearning setting, across fractions of identified manipulated samples $\frac{|S_f|}{|S_m|}$. We investigate the unlearning of two manipulations: poisoning (Gu et al., 2019) and interclass confusion (Goel et al., 2023).

Roadmap: We report the Experimental Setup in Section 3 with further details in Appendix Section B. Table 1 lists the quantities reported on the Y-axis to measure removal (1) and utility (2). To measure effectiveness at different levels of identification of manipulated samples (3), we vary $|S_f|$ on the X-axis from 10% of $|S_m|$, i.e. a small portion of manipulated samples being used for unlearning, to 100% of $|S_m|$, i.e. all manipulated samples being used for unlearning. Finally, we report computational efficiency (4) of the different methods used in Table 3.

Setup Details

Datasets, Models, Manipulation and Deletion Sizes: We use the CIFAR (Krizhevsky et al., 2009) datasets as standard benchmarking datasets in image classification and unlearning. We use the ResNet-9 (Idelbayev, 2018) model for CIFAR10, and WideResNet-28x10 (Zagoruyko & Komodakis, 2016) for CIFAR100. We report results for each dataset for multiple manipulation sizes $n = |S_m|$ as detailed in Table 2. In each setting, we vary the deletion set size $|S_f|$ from 10% to 100% of the manipulation size $|S_m|$ at intervals of 10%.

Unlearning Methods: We benchmark state-of-the-art unlearning methods. Detailed descriptions, and specification of the hyperparameter search across all methods is provided in Appendix Section B.

(1) Exact Unlearning (EU): EU retrains from scratch on $S_{tr} \setminus S_f$ using the original training algorithm A. This is considered an inefficient but gold-standard oracle in the unlearning literature. Many existing methods are weaker relaxations of this procedure (He et al., 2021; Graves et al., 2021; Goel et al., 2023).

(2) Catastrophic Forgetting (CF): Goel et al. (2023) shows that finetuning just the final layers of a deep model on $S_{tr} \setminus S_f$ performs well at unlearning label manipulations. We use the strongest version of this by using all layers for unlearning.

(3) Selective Synaptic Dampening (SSD): Foster et al. (2023) selectively modifies learnt weights with high influence from S_f, which are identified by approximating the Fisher Information Matrix (Martens, 2020).

(4) Knowledge Distillation from a Bad Teacher (BadT): Chundawat et al. (2023b) proposes making outputs on S_f random by distilling from a randomly initialized network. To retain utility, they simultaneously distill from the original model on $S_{tr} \setminus S_f$.

(5) SCalable Remembering and Unlearning unBound (SCRUB): Kurmanji et al. (2023) proposes alternating between steps of gradient ascent on S_f and knowledge preservation on $S_{tr} \setminus S_f$ using distillation from the original model along with a task-specific loss.

Unlearning Poisons

Setting: We use the BadNet poisoning attack introduced by Gu et al. (2019) to evaluate the use of unlearning methods to remove backdoors. We manipulate n training images by inserting a trigger pattern that makes 0.3% pixels white at bottom-right positions, re-labeling each of these images to

Dataset	#Classes	Model	Poisoning $|S_m|/|S_{tr}|$	IC Test $|S_m|/|S_{tr}|$
CIFAR-10	10	ResNet-9	0.2%, 1%, 2%	1%, 5%, 10%
CIFAR-100	100	WideResNet-28x10	0.2%, 1%, 2%	0.2%, 0.5%, 1%

Table 2: Dataset and models along with manipulation sizes for the Poisoning and Interclass Confusion (IC) evaluation.

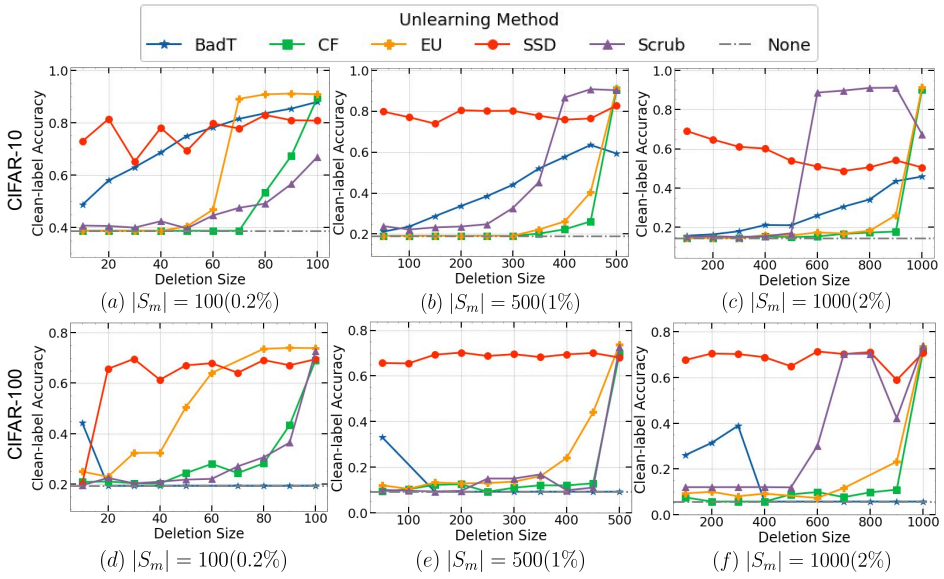

Figure 2: **Clean-label Accuracy on Test Samples with Poison Trigger**. Each method is shown across deletion sizes $|S_f|$ after unlearning ("None" represents the original model). Existing unlearning methods except SSD, including EU which is traditionally considered a gold-standard, perform poorly when $\leq 80\%$ of the poisoned data is identified for unlearning, even when just 1% of training data is poisoned as in (b), (e).

class zero. Models trained on datasets containing this manipulation are more likely to label samples containing the trigger pattern as class zero. Here the affected domain $\mathcal{D}_m$ consists of all samples containing the trigger pattern. In this setting, adversaries manipulate both the data features and labels. This can occur when model developers scrape data and corresponding annotations from webpages, such that a subset of these webpages can be manipulated by the adversary.

Results:Figure 2 shows clean-label accuracies when the trigger pattern is inserted in all test set samples. EU is the gold standard when all manipulated samples are known, and indeed it achieves the highest accuracy at $|S_f| = |S_m|$. However, it dramatically fails in cases when up to 80% of the manipulated samples are known, even where only 1% (500 samples) of the training data is poisoned (subfigures b, e). This shows the insufficiency of the traditional unlearning goal of approximating retraining from scratch on $S_{tr} \setminus S_f$, as the remaining poisoned samples are capable of maintaining their adverse effects, even when their number is small (Gu et al., 2019).

As a consequence, state-of-the-art approaches in unlearning literature like EU, CF, and Scrub perform quite poorly in this setting. BadT shows poor results throughout, as randomizing outputs on S_f conflicts with the goal of improving model accuracy on S_f (1). On the contrary, SSD recovers accuracy on $\mathcal{D}_m$ (achieving 1) even with 10% of manipulated samples known, showing the tractability of generalizing removal from a small representative subset of S_m (3). However, as shown in Figure 7 (Appendix), SSD leads to significant drops in model utility (2), while other unlearning methods maintain utility throughout. Pruning a small subset of weights is a well-known strategy to mitigate poisons (Wang et al., 2019) as they associate a specific feature with the incorrect

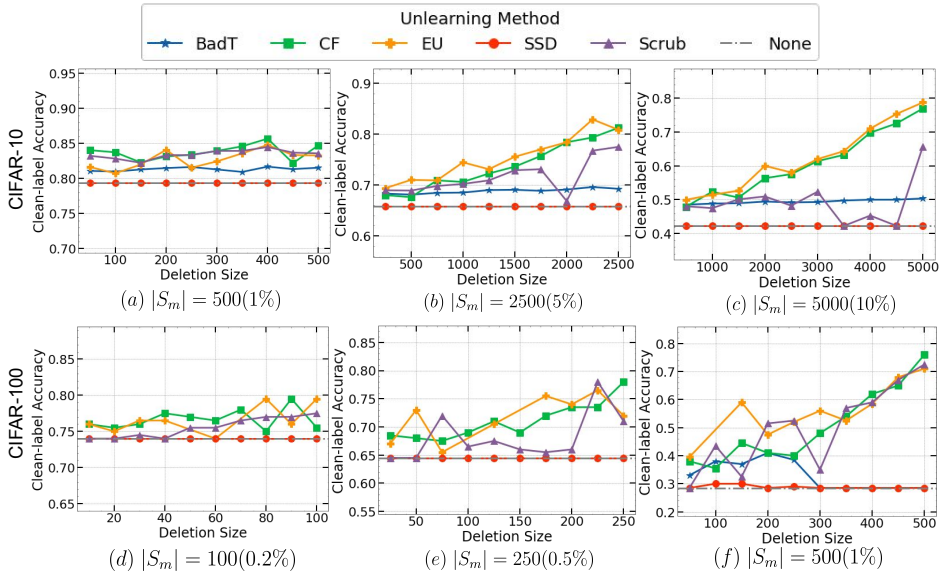

Figure 3: **Clean-label Accuracy on Test Samples on the Two Confused Classes**. We compute clean-label accuracy on the classes A, B used for the Interclass Confusion test, across deletion sizes $|S_f|$. SSD provides no improvements over the original model (represented as "None"), and other unlearning methods also require a large fraction of the manipulated data to be identified for unlearning. In the lower manipulation size setting (a) and (d), the model outputs on unseen samples are not affected much, so we show unlearning trends on manipulated train samples below.

label. We believe SSD succeeds in this setting as it can identify weights that learn the BadNet poison effectively even when only a small portion of the manipulation set is known.

Conclusion: Traditional unlearning methods that train on $S_{tr} \setminus S_f$ perform poorly in practical scenarios when all manipulated samples are unknown (3). SSD shows positive results for removing poisons, demonstrating the tractability of corrective unlearning in this setting, though it hurts model utility, leaving scope for improvements. Since SSD works by modifying a small subset of weights, it motivates the usefulness of mechanistic interpretability (Elhage et al., 2021) or influence-function based approaches (Grosse et al., 2023) for removing backdoors at least in small-scale settings.

Unlearning Interclass Confusion

Setting: We use the Interclass Confusion (IC) test as a strong evaluation for the use of unlearning methods to remove the influence of mislabels. In the IC test, two classes A and B are picked, and $\frac{n}{2}$ samples from both classes are selected, and their label is changed to the other class. Models trained on datasets containing this manipulation are more likely to confuse these classes, i.e. predict A samples as B and vice-versa. The affected domain $\mathcal{D}_m$ consists of all samples from class A and class B. For CIFAR10, we confuse the Cat and Dog classes, and for CIFAR100 maple and oak tree, which is consistent with the setup of Goel et al. (2023).

The IC test applies in the setting where the adversary can only manipulate labels, such as when model developers outsource annotations for their own data. Mislabels between two classes can also occur due to systematic biases in the labelling process, or misinterpretation in annotation guidelines on how to distinguish the classes. Manipulating only labels may appear to be a weaker setting compared to poisoning. However, unlike poisoning where a small subset of weights may be associated with the trigger and can be targeted for unlearning, the IC test can have a more uniform effect across weights, confusing the learnt representations of clean samples without any specific triggers. We hypothesize unlearning procedures like SSD that modify specific parameters may be less effective for such settings.

Results: In Figure 3, we see that EU, CF, and Scrub show gradual improvement in removal (1) as larger fractions of the manipulated set are identified. BadT performs poorly across deletion set sizes, similar to poisoning. While SSD, a mechanistic intervention that prunes certain weights, showed promising results for poison removal, it completely fails at removing interclass confusion.

Conclusion: The failure of SSD in this setting highlights the need for evaluating unlearning procedures with diverse manipulations. Traditional unlearning procedures have poor removal (1) when small subsets of the manipulation set are identified (3). Overall, there is scope for designing better corrective unlearning methods that achieve desiderata 1-3 across different manipulation types.

4 Future Work

The ideal corrective unlearning approaches should exhibit robustness against a broad spectrum of manipulation types. Specifically, these methods should withstand adaptive attacks, where the manipulations targeted for unlearning are crafted with knowledge of the unlearning procedures themselves (Tramer et al., 2020), not just the two evaluations we study. Similar to other related fields like adversarial robustness and privacy, it is important to design new Corrective Unlearning algorithms that work against powerful adaptive attacks.

In addition, there is scope to design stronger evaluation frameworks for corrective unlearning. Apart from manipulating features and labels, adversaries could generate entirely synthetic samples (Zhang et al., 2019; Huang et al., 2020). Although our focus is on supervised image classification, the concept of manipulation and its correction is also relevant in self-supervised learning contexts, such as language modeling (Wallace et al., 2020). Finally, an additional complexity could be the presence of false positives, where a clean sample getting identified as manipulated.

Current unlearning procedures aim to achieve a model distribution that is indistinguishably close to one obtained by retraining without certain samples, measured in terms like (ϵ, δ)-certified unlearning (Sekhari et al., 2021). However, we anticipate that the corrective unlearning problem will pave the way for innovative theoretical research. A critical area of interest is determining what conditions make a small 'representative set' of manipulated samples sufficient for effective corrective unlearning. Additionally, for a given manipulation class and a small set of such samples, it would be interesting to develop algorithms that prioritize improving accuracy on the manipulated domain over strict distributional indistinguishability. Another future challenge is to identify additional manipulated samples based on a small initial representative set.

5 Conclusion

Overall, we explore the Corrective Machine Unlearning setting, designed to mitigate the negative effects of manipulated data discovered post-training, such as diminished accuracy across specific domain areas, from an already trained model. This concept is grounded in an adversarial threat model, acknowledging that all the manipulated data samples may not be known. Instead, developers are often able to pinpoint only a representative subset of the manipulated samples.

Our setting diverges significantly from the traditional unlearning setting, which is primarily designed to address privacy concerns. Our findings indicate that latest unlearning methods, even the gold standard of retraining-from-scratch, fail to enhance accuracy on the manipulated domain unless nearly all of the manipulated data is identified. A notable exception is SSD (Foster et al., 2023), which successfully mitigates the effects of the BadNet (Gu et al., 2019) poison, thus illustrating the feasibility of removing the influence of manipulated data with only a small representative subset identified. However, this method does not work for the Interclass Confusion (Goel et al., 2023) manipulation, which demonstrates the need for designing unlearning procedures that can ideally remove the influence of arbitrary manipulations. We hope our work spurs the development of stronger corrective unlearning methods and evaluations to assist practitioners in dealing with data quality issues arising from web-scale training.

ACKNOWLEDGEMENTS

SG is funded by an Effective Ventures Foundation (UK) grant as part of the ML Alignment Theory Scholars (MATS) program. AP is funded by Meta AI Grant No. DFR05540. PT thanks the Royal Academy of Engineering and FiveAI for their support. This work is supported in part by a UKRI grant: Turing AI Fellowship EP/W002981/1 and an EPSRC/MURI grant: EP/N019474/1. The authors would like to thank (in alphabetical order): Arvindh Arun, Shyamgopal Karthik, Shashwat Singh, Shiven Sinha, Vishaal Udandarao, Christian Schroeder de Witt for helpful feedback, and Varshita Kolipaka for contributing illustrations.

REFERENCES

Marco Barreno, Blaine Nelson, Russell Sears, Anthony D Joseph, and J Doug Tygar. Can machine learning be secure? In *ASIA Conference on Computer and Communications Security (ACM ASIACCS)*, 2006.

David Bau, Steven Liu, Tongzhou Wang, Jun-Yan Zhu, and Antonio Torralba. Rewriting a deep generative model. In *European Conference on Computer Vision*, 2020.

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

Eric Breck, Marty Zinkevich, Neoklis Polyzotis, Steven Whang, and Sudip Roy. Data validation for machine learning. In *Proceedings of SysML*, 2019.

Carla E Brodley and Mark A Friedl. Identifying mislabeled training data. *Journal of Artifical Intelligence Research (JAIR)*, 1999.

California State Leglisature. California consumer privacy act, 2018.

Nicholas Carlini, Matthew Jagielski, Christopher A. Choquette-Choo, Daniel Paleka, Will Pearce, Hyrum Anderson, Andreas Terzis, Kurt Thomas, and Florian Tramèr. Poisoning web-scale training datasets is practical. In *IEEE Symposium on Security and Privacy (SP)*, 2023.

Huili Chen, Cheng Fu, Jishen Zhao, and Farinaz Koushanfar. Deepinspect: A black-box trojan detection and mitigation framework for deep neural networks. In *International Joint Conference on Artificial Intelligence*, 2019.

Jiaao Chen and Diyi Yang. Unlearn what you want to forget: Efficient unlearning for LLMs. In *Conference on Empirical Methods in Natural Language Processing*, 2023.

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

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 *Annual AAAI Conference on Artificial Intelligence*, 2023b.

Council of European Union. Eu general data protection regulation, 2018.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography (TOC*, 2006.

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

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021.

Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. *Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

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

Robert M. French. Catastrophic forgetting in connectionist networks. In *Trends in Cognitive Sciences*, 1999.

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

Antonio Ginart, Melody Y. Guan, Gregory Valiant, and James Zou. Making AI forget you: Data deletion in machine learning. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

Shashwat Goel, Ameya Prabhu, Amartya Sanyal, Ser-Nam Lim, Philip Torr, and Ponnurangam Kumaraguru. Towards adversarial evaluations for inexact machine unlearning. *arXiv:2201.06640*, 2023.

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020a.

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Forgetting outside the box: Scrubbing deep networks of information accessible from input-output observations. In *European Conference on Computer Vision*, 2020b.

Micah Goldblum, Dimitris Tsipras, Chulin Xie, Xinyun Chen, Avi Schwarzschild, Dawn Song, Aleksander Madry, Bo Li, and Tom Goldstein. Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

Laura Graves, Vineel Nagisetty, and Vijay Ganesh. Amnesiac machine learning. In *Annual AAAI Conference on Artificial Intelligence*, 2021.

Roger Grosse, Juhan Bae, Cem Anil, Nelson Elhage, Alex Tamkin, Amirhossein Tajdini, Benoit Steiner, Dustin Li, Esin Durmus, Ethan Perez, et al. Studying large language model generalization with influence functions. *arXiv:2308.03296*, 2023.

Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 2019.

Varun Gupta, Christopher Jung, Seth Neel, Aaron Roth, Saeed Sharifi-Malvajerdi, and Chris Waites. Adaptive machine unlearning. *Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

Xingshuo Han, Guowen Xu, Yuan Zhou, Xuehuan Yang, Jiwei Li, and Tianwei Zhang. Physical backdoor attacks to lane detection systems in autonomous driving. In *ACM International Conference on Multimedia*, 2022.

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

W Ronny Huang, Jonas Geiping, Liam Fowl, Gavin Taylor, and Tom Goldstein. Metapoison: Practical general-purpose clean-label data poisoning. *Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

Yerlan Idelbayev. Proper ResNet implementation for CIFAR10/CIFAR100 in PyTorch, 2018.

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

Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. Knowledge unlearning for mitigating privacy risks in language models. In *Annual Meeting of the Association for Computational Linguistics*, 2023.

Heinrich Jiang and Ofir Nachum. Identifying and correcting label bias in machine learning. In *International Conference on Artificial Intelligence and Statistics*, 2020.

Nikola H Konstantinov and Christoph Lampert. Fairness-aware pac learning from corrupted data. *Journal of Machine Learning Research (JMLR)*, 2022.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009.

Meghdad Kurmanji, Peter Triantafillou, and Eleni Triantafillou. Towards unbounded machine unlearning. *Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

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

Zhuo Ma, Yang Liu, Ximeng Liu, Jian Liu, Jianfeng Ma, and Kui Ren. Learn to forget: Machine unlearning via neuron masking. *IEEE Transactions on Dependable and Secure Computing*, 2023.

James Martens. New insights and perspectives on the natural gradient method. *Journal of Machine Learning Research (JMLR)*, 2020.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. Memory-based model editing at scale. In *International Conference on Machine Learning (ICML)*, 2022.

Seth Neel, Aaron Roth, and Saeed Sharifi-Malvajerdi. Descent-to-delete: Gradient-based methods for machine unlearning. In *Conference on Learning Theory (COLT)*, 2021.

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

Curtis G. Northcutt, Anish Athalye, and Jonas Mueller. Pervasive label errors in test sets destabilize machine learning benchmarks. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021a.

Curtis G. Northcutt, Lu Jiang, and Isaac L. Chuang. Confident learning: Estimating uncertainty in dataset labels. In *Journal of Artifical Intelligence Research (JAIR)*, 2021b.

Daniel Paleka and Amartya Sanyal. A law of adversarial risk, interpolation, and label noise. In *International Conference on Learning Representations (ICLR)*, 2023.

Parliament of Canada. Personal information protection and electronic documents act, 2018.

Alexandra Peste, Dan Alistarh, and Christoph H Lampert. Ssse: Efficiently erasing samples from trained machine learning models. In *NeurIPS Workshop Privacy in Machine Learning*, 2021.

Geoff Pleiss, Tianyi Zhang, Ethan Elenberg, and Kilian Q Weinberger. Identifying mislabeled data using the area under the margin ranking. *Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

Vinay Uday Prabhu and Abeba Birhane. Large image datasets: A pyrrhic win for computer vision? In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)* , 2021.

Amartya Sanyal, Puneet K. Dokania, Varun Kanade, and Philip Torr. How benign is benign overfitting? In *International Conference on Learning Representations (ICLR)*, 2021.

Amartya Sanyal, Yaxi Hu, and Fanny Yang. How unfair is private learning? In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2022.

Sebastian Schelter. "amnesia" - machine learning models that can forget user data very fast. In *Conference on Innovative Data Systems Research (CIDR)*, 2020.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

Ayush Sekhari, Jayadev Acharya, Gautam Kamath, and Ananda Theertha Suresh. Remember what you want to forget: Algorithms for machine unlearning. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *IEEE Symposium on Security and Privacy (SP)*, 2017.

David M. Sommer, Liwei SOng, Sameer Wagh, and Prateek Mittal. Athena: Probabilistic Verification of Machine Unlearning. *Proceedings on Privacy Enhancing Technologies (PoPETS)*, 2022.

Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

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

Anvith Thudi, Hengrui Jia, Ilia Shumailov, and Nicolas Papernot. On the necessity of auditable algorithmic definitions for machine unlearning. In *USENIX*, 2022.

Zhiyi Tian, Lei Cui, Jie Liang, and Shui Yu. A comprehensive survey on poisoning attacks and countermeasures in machine learning. *ACM Computing Surveys*, 2022.

Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. *Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

Eric Wallace, Tony Z Zhao, Shi Feng, and Sameer Singh. Concealed data poisoning attacks on nlp models. *arXiv:2010.12563*, 2020.

Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y. Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *IEEE Symposium on Security and Privacy (SP)*, 2019.

Alexander Warnecke, Lukas Pirch, Christian Wressnegger, and Konrad Rieck. Machine unlearning of features and labels. *Network and Distributed System Security Symposium*, 2021.

Yinjun Wu, Edgar Dobriban, and Susan B. Davidson. Deltagrad: Rapid retraining of machine learning models. In *International Conference on Machine Learning (ICML)*, 2020.

Yuanshun Yao, Xiaojun Xu, and Yang Liu. Large language model unlearning. *arXiv:2310.10683*, 2023.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *The British Machine Vision Conference*, 2016.

Jiale Zhang, Junjun Chen, Di Wu, Bing Chen, and Shui Yu. Poisoning attack in federated learning using generative adversarial nets. In *IEEE international conference on big data science and engineering (TrustCom/BigDataSE)*, 2019.

A Related Work

Learning from manipulated data: The adverse effects of manipulated training data on machine learning models are well-documented across objectives like fairness (Konstantinov & Lampert, 2022), robustness (Sanyal et al., 2021; Paleka & Sanyal, 2023), and adversarial reliability (Tian et al., 2022). One line of defense is designing training strategies more robust to these issues, see Song et al. (2022) for a survey on learning with mislabels. However, learning robust models from manipulated data is a hard problem as reduced sensitivity to such minority data populations can harm accuracy and fairness (Feldman & Zhang, 2020; Sanyal et al., 2022). Unlearning specific samples which are discovered to be manipulated can be a complementary mitigation approach. Further, we hope corrective unlearning procedures are compared using the same original model, to ensure improvements are due to the unlearning procedure rather than properties of the original training procedure or model.

How to detect manipulated data? A prerequisite to the corrective unlearning task is detecting a representative subset of manipulated data. Fortunately, this has long been studied (Brodley & Friedl, 1999), with prior work detailing techniques to discover mislabeled (Pleiss et al., 2020; Northcutt et al., 2021a), biased (Prabhu & Birhane, 2021; Jiang & Nachum, 2020) and poisoned (Chen et al., 2019; Wang et al., 2019) data. Further, compromised data sources can be identified using web security and data collection practices. We assume the model developers employ such strategies for monitoring their data sources. However, they cannot simply throw away the trained model when manipulated data is found due to expensive retraining costs. We study how to cheaply mitigate adverse effects on such models using unlearning.

Known Manipulations or Correct Labels: If the type of manipulation is known, one may employ manipulation-specific mitigation techniques such as poisoning (sometimes referred to as trojan) defences (see Goldblum et al. (2022) for a survey). We restrict the scope of our work to not knowing the precise manipulation, and study the use of unlearning as a broader panacea procedure across unknown data manipulations. Finally, if the samples can be corrected through re-annotation, one may also use knowledge editing techniques (Bau et al., 2020; Mitchell et al., 2022).

Unlearning: Prior work in designing unlearning procedures is motivated by privacy applications, and aims to achieve *retrain indistinguishability* (Ginart et al., 2019; Golatkar et al., 2020a), that is to create a distribution of unlearnt models indistinguishable from retraining from scratch without the data to be deleted. In Section 2 we discuss differences in corrective unlearning desiderata from retrain indistinguishability. "Exact Unlearning" procedures ensure the unlearnt model never sees the data whose influence is to be deleted by design of the training procedure (Bourtoule et al., 2021; Schelter, 2020). The empirical results of EU in Section 3 show how these approaches may not suffice for corrective unlearning when the full manipulation set is unknown. Moreover, such methods drastically deteriorate in efficiency as the as the number of samples to delete increase (Warnecke et al., 2021). This has led to "Inexact Unlearning" proposals, and we use state of art methods in image classification from different paradigms for our experiments:

- Modifying parameters which influence forget set outputs (Golatkar et al., 2020a; Peste et al., 2021; Ma et al., 2023) - We benchmark Selective Synaptic Dampening (SSD) (Foster et al., 2023).
- Randomizing model outputs on the data to be deleted (Graves et al., 2021; Chundawat et al., 2023a; Tarun et al., 2023) - We benchmark Knowledge Distillation from Bad Teacher (BadT) (Chundawat et al., 2023b).
- Finetuning based approaches only using retained samples (Warnecke et al., 2021; Yao et al., 2023; Jang et al., 2023; Eldan & Russinovich, 2023; Chen & Yang, 2023) - We benchmark Catastrophic Forgetting (CF), as Goel et al. (2023) show it works well on the Interclass Confusion test.
- Alternating between Forgetting and Preservation Steps - We use SCRUB as Kurmanji et al. (2023) show it works well on the Interclass Confusion test.

A group of works (Izzo et al., 2021; Wu et al., 2020; Gupta et al., 2021; Neel et al., 2021; Thudi et al., 2022; Sekhari et al., 2021) also study unlearning procedures on convex or linear models with theoretical guarantees inspired from differential privacy (Dwork et al., 2006), but in this work we focus on deep models. Finally, Goel et al. (2023); Kurmanji et al. (2023); Sommer et al. (2022) consider unlearning of mislabelled or poisoned samples, but only as a stronger evaluation for the

privacy-oriented objective of retrain indistinguishability. We show retraining cannot be used as a gold standard for corrective unlearning when only a subset of manipulated samples is identified (3), which leads to the insufficiency of unlearning methods geared towards indistinguishability from retraining for corrective unlearning.

B Experimental Setup Details

Training Details

Our standard training procedure A is as follows: We train our models for 4000 steps on CIFAR10, PCAM and 6000 steps on CIFAR100. Each step consists of training on a single batch, and we use a batch size of 512 throughout. We use an SGD optimizer with momentum 0.9 and weight decay 5e-4, a linear scheduler with t_{mult} = 1.25, and warmup steps as $\frac{1}{100}$ of the total training steps. The same hyperparameters are used during unlearning unless otherwise specified. The setup used for all experiments is a PC with a Intel(R) Xeon(R) E5-2640 2.40 GHz CPU, 128GB RAM and 1 GeForce RTX 2080 GPU.

Detailed Description of Unlearning Methods

To benchmark the performance of existing unlearning proposals on corrective unlearning scenarios, we select the strongest unlearning methods across five popular paradigms:

(1) Exact Unlearning (EU): This paradigm involves retraining parts of the ML system (Bourtoule et al., 2021; Goel et al., 2023; He et al., 2021) that are influenced by S_f from scratch using $S_{tr} \setminus S_f$.

Method Used: We benchmark the strongest version, retraining the entire model from scratch on $S_{tr} \setminus S_f$ using the original training algorithm A. This is considered an inefficient but gold standard unlearning procedure in prior work.

(2) Catastrophic Forgetting (CF) : Neural Networks suffer from catastrophic-forgetting (French, 1999) - when a model is continually updated without some previously learnt samples, the model loses knowledge about them. Many unlearning methods perform finetuning on $S_{tr} \setminus S_f$ to achieve unlearning of S_f via catastrophic forgetting, and Goel et al. (2023) show even finetuning just the final layers of the model performs well on the IC test.

Method Used: We use the original training procedure A for 1000 steps on $S_{tr} \setminus S_f$.

(3) Modifying learnt parameters with high influence from S_f: This is a training-free class of methods (Golatkar et al., 2020a;b; Peste et al., 2021; Chundawat et al., 2023a) that identifies parameters with information relevant to the forget set using statistics like the Fisher Information Matrix (FIM). It then damages these parameters by adding noise or reducing their magnitude hoping to selectively remove information about S_f.

Method Used: We benchmark the recently proposed Selective Synaptic Dampening (SSD) method which has shown state of the art results in this paradigm (Foster et al., 2023). We extensively tune the weight selection threshold α and weight dampening constant γ. We find that γ should be tuned relative to α for optimal results. For each datapoint, we pick the best result out of runs with $\alpha = [0.1, 1, 10, 50, 100, 500, 1000, 1e4, 1e5, 1e6]$, $\gamma = [0.1\alpha, 0.5\alpha, \alpha, 5\alpha, 10\alpha]$.

(4) Pushing S_f outputs towards random: Some unlearning procedures (Graves et al., 2021; Li & Ghosh, 2023; Chundawat et al., 2023b) push the model towards random outputs on the deletion set.

Method Used: We benchmark Knowledge Distillation from Bad Teacher (BadT) (Chundawat et al., 2023b), a state of the art method in this paradigm, which simultaneously distills from a randomly initialized neural network on S_f, and the original model on the remaining data $S_{tr} \setminus S_f$. We finetune the original model using this procedure for 1000 unlearning steps.

(5) Alternating between Forgetting and Preservation Steps:

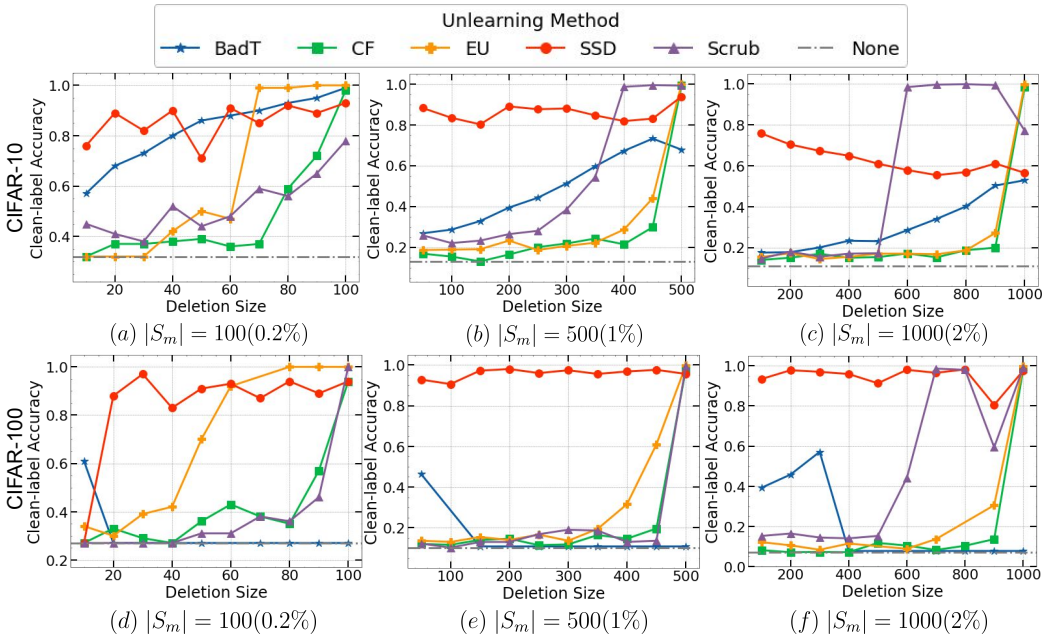

Figure 4: **Clean-label Accuracy on Manipulated Train Samples S_m with Poison Trigger.** Each method is shown across deletion sizes $|S_f|$ after training with adversarial poisoning ("None" represents the original model). Trends mimic results for clean-label accuracy on unseen samples with the poison trigger.

Method Used: Kurmanji et al. (2023) propose SCRUB and show it performs well on unlearning mislabelled samples when all are identified. The method alternates between forget steps and knowledge preservation steps. The forget step involves doing gradient ascent using the task-loss for S_f. The knowledge preservation step does knowledge distillation from M_o using $S_{tr} \setminus S_f$ as well as optimizing the task-loss on $S_{tr} \setminus S_f$. We finetune the original model using this procedure for 1000 unlearning steps, out of which the forget step is used only in the first 200 unlearning steps as it is recommended in the paper to run it only in the initial iterations. We use a smaller learning rate (0.0025) as the original value leads to stability issues. We tune the hyperparameter α which controls the trade-off between the distillation loss and the task-loss. For each datapoint, we pick the best result out of runs with $\alpha = [0.001, 0.01, 0.05, 0.1, 0.5, 1, 5, 10]$.

Selection of Best Hyperparameters in Unlearning Phase

Most unlearning methods require hyperparameter tuning and this presents a challenge for the model developers on how to pick the best model. Selecting the model with the best validation accuracy may have low removal (❶), especially if the domain affected by the manipulation $\mathcal{D}_m$ is a small fraction of the overall domain $\mathcal{X}$. Moreover, model developers are unaware of the manipulation performed by an adversary, and thus may not be able to precisely isolate the affected domain for validation. In our setting, model developers only have access to S_f; thus even assuming the original training to be incorrect, the correct labels are unknown in multiclass setting. Let the *deletion change* be the fraction of S_f whose prediction by the model differs from the provided label in training. A higher deletion change may indicate more removal. However, note that the deletion change of a trivial model that has no utility (❷) can be quite high. Thus, we propose using **a weighted average of the deletion change and the validation accuracy** to select an unlearnt model that balances removal (❶) and utility (❷). In this work, we weigh them equally.

C Further Results

We now provide results not included in the main paper to ensure completeness. Specifically:

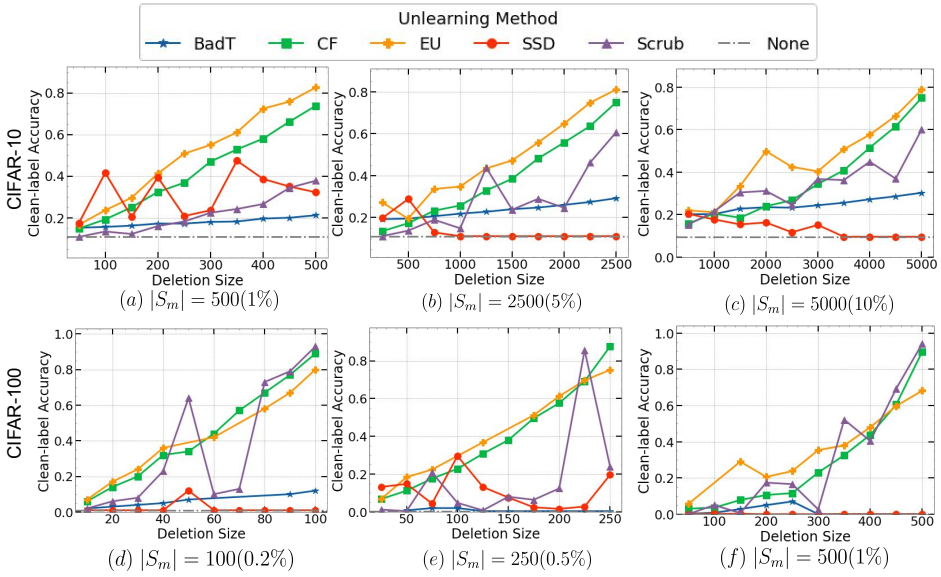

Figure 5: **Clean-label Accuracy on Manipulated Training Samples S_m with Interclass Confusion** for different unlearning methods ("None" represents the original model) across deletion sizes $|S_f|$. Existing unlearning methods perform poorly when $\frac{|S_f|}{|S_m|}$ is lower. Even the smallest setting (a, d) shows clear unlearning trends.

- We report clean-label accuracies on manipulated training samples.
- We report utility of models after unlearning
- We report computational efficiency by measuring unlearning time for each method.

Removal Measured on Manipulated Training Samples

To measure the removal of mislabelling on poisoned training samples, we report clean-label accuracy on S_m in Figure 4. The trends across unlearning methods are similar to the ones on unseen samples from the affected domain $\mathcal{D}_m$ reported in the main paper, though the absolute accuracies after unlearning are higher as expected from training samples in comparison to test set samples.

Finally, while the smallest manipulation size (subfigures a, d) for Interclass Confusion did not show significant effects on unseen samples from class A, B, figure 5 shows unlearning methods continue to give wrong predictions on the class A, B samples used for training. This emphasises the need to check unlearnt model outputs on unseen training samples from the affected domain $\mathcal{D}_m$ in addition to test samples from $\mathcal{D}_m$.

Utility after Removal

In Figure 7 we see that SSD leads to significant drops in test accuracy on clean (not poisoned) samples, while other methods maintain utility.

In Figure 6 we plot the utilities across deletion set sizes for IC test. We report accuracies on unseen samples from the classes not manipulated by interclass confusion. These samples can be considered to belong to the same distribution as $S_{tr} \setminus S_m$. We find methods maintain accuracy, and EU, CF even show minor (0.5-1%) gains when most of the manipulated data is known. This is not surprising as removing the effect of manipulations can improve learnt representations and the overall utility of the model.

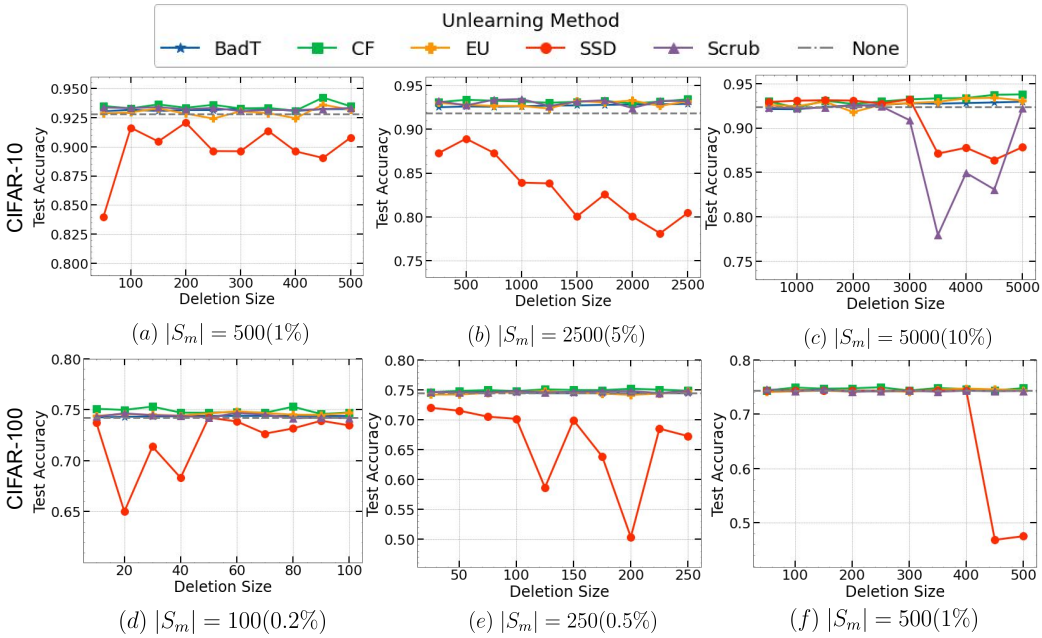

Figure 6: **Accuracy on Test Samples from classes other than the two confused.** Except SSD which shows drops in utility, we see similar accuracies across different unlearning methods across deletion sizes $|S_f|$ after training with Interclass Confusion ("None" represents the original model).

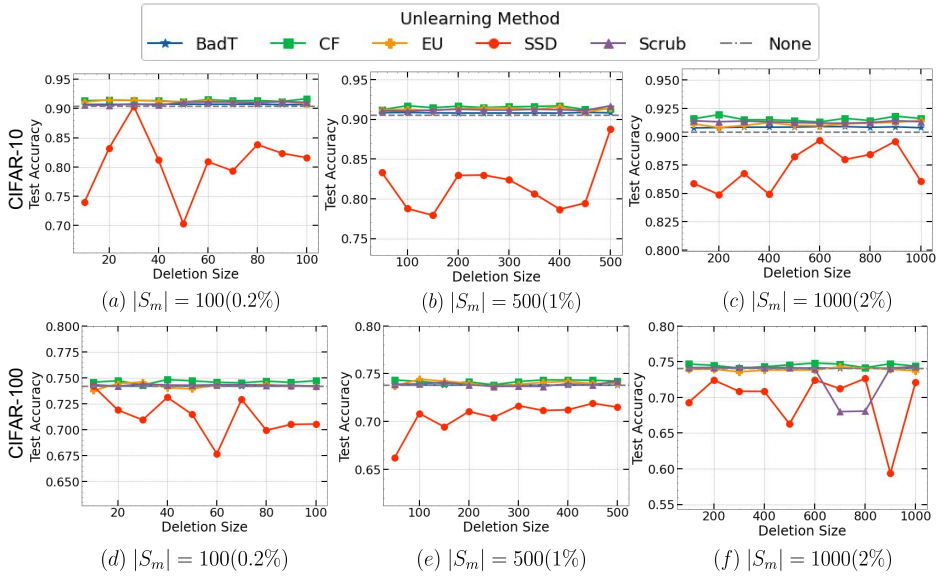

Figure 7: **Accuracy on Test Samples with No Poison trigger**. While other unlearning methods ("None" represents the original model) maintain utility, SSD shows a significant drop across deletion sizes $|S_f|$ across (a)-(f).

Computational Efficiency

In Table 3 we report average unlearning times of different unlearning methods. In the case of EU and CF, while more efficient relaxations have been proposed (Goel et al., 2023; He et al., 2021; Graves et al., 2021), we retrain from scratch to perform the strongest unlearning, which we still find to be insufficient.

Method	**Time** (minutes)
EU	49.93
CF	10.52
Scrub	16.86
SSD	1.80
BadT	33.19

Table 3: Unlearning Time by Method