Δ-Influence: Unlearning Poisons via Influence Functions

Wenjie Li
ShanghaiTech University

Jiawei Li
Tsinghua University

Christian Schroeder de Witt
University of Oxford

Ameya Prabhu
Tübingen AI Center, University of Tübingen

Amartya Sanyal
University of Copenhagen

Abstract

Addressing data integrity challenges, such as unlearning the effects of data poisoning after model training, is necessary for the reliable deployment of machine learning models. State-of-the-art influence functions, such as EK-FAC [Grosse et al., 2023], often fail to accurately attribute abnormal model behavior to the specific poisoned training data responsible for the data poisoning attack. In addition, traditional unlearning algorithms often struggle to effectively remove the influence of poisoned samples [Pawelczyk et al., 2024], particularly when only a few affected examples can be identified [Goel et al., 2024]. To address these challenge, we introduce Δ-Influence, a novel approach that leverages influence functions to trace abnormal model behavior back to the responsible poisoned training data using as little as just one *one* poisoned test example. Δ-Influence applies data transformations that sever the link between poisoned training data and compromised test points without significantly affecting clean data. This allows Δ-Influence to detect large negative shifts in influence scores following data transformations, a phenomenon we term as influence collapse, thereby accurately identifying poisoned training data. Unlearning this subset, e.g. through retraining, effectively eliminates the data poisoning. We validate our method across three vision-based poisoning attacks and three datasets, benchmarking against four detection algorithms and five unlearning strategies. We show that Δ-Influence consistently achieves the best unlearning across all settings, showing the promise of influence functions for corrective unlearning. Our code is publicly available at: `https://github.com/andyisokay/delta-influence`

1 Introduction

Machine learning models are increasingly deployed in critical sectors such as healthcare, finance, and autonomous systems [Chatila et al., 2021, Davenport and Kalakota, 2019, Huang et al., 2020, Soori et al., 2023]. This widespread adoption underscores the importance of ensuring model integrity and robustness against malicious attacks, particularly data poisoning attacks. In data poisoning, adversaries intentionally manipulate training data by introducing carefully crafted, often imperceptible modifications [Chatila et al., 2021], leading to incorrect predictions or embedding specific malicious behaviors within the trained models [Fan et al., 2022]. Given the large scale of modern datasets, identifying and removing all manipulated samples is typically impractical [Goel et al., 2024, Nguyen et al., 2024a]. Therefore, a viable approach involves detecting and attributing the impact of data poisoning to a small set of *influential* training data points, which is *unlearned* to mitigate the data poisoning attack.

The challenge of effective unlearning largely depends on the extent of knowledge about the data poisoning attack. For example, Goel et al. [2024] demonstrate that retraining a model after removing

Presented at Workshop on Attributing Model Behavior at Scale (ATTRIB @ NeurIPS 2024).

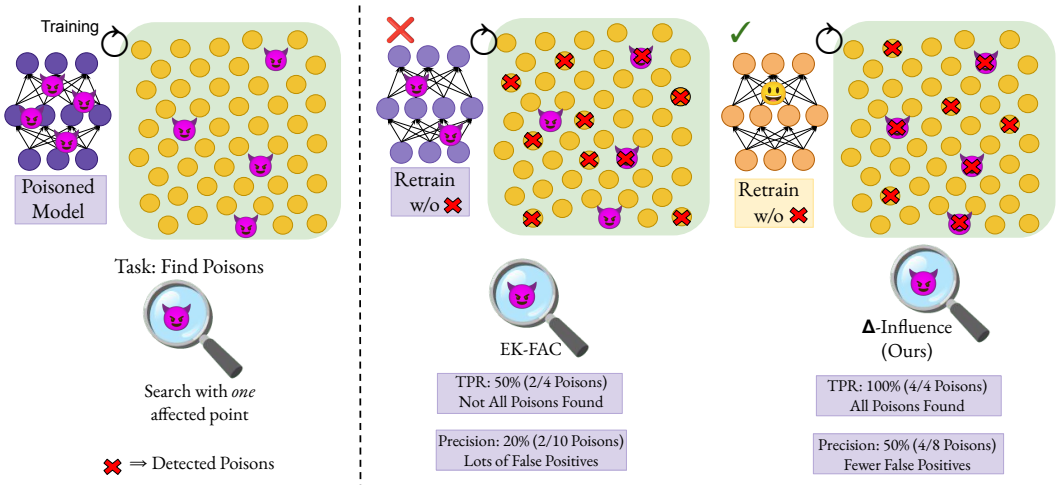

Figure 1: Given an affected test point, our goal is to identify the training points responsible for the poisoning, so that retraining without these points can remove the attack from the model. State-of-the-art methods like EK-FAC [Grosse et al., 2023] detect only a few poisoned points with low precision, leaving the poisoning effect in the model and causing a large accuracy drop. Our method, $\Delta - \text{Influence}$, outperforms existing approaches by successfully recovering the clean model without sacrificing accuracy.

a randomly sampled subset containing half of the manipulated data fails to eliminate poisoning in relatively simple attacks like BadNet [Gu et al., 2019]. In contrast, retraining without the entire set of manipulated data successfully removes the attack, highlighting the limitations of partially identifying manipulated data. Furthermore, for more sophisticated poisoning strategies such as Witches' Brew [Geiping et al., 2021], Pawelczyk et al. [2024] reveal that existing unlearning algorithms are ineffective unless the model is retrained without the full manipulated set, even when full access to the manipulated data is available.

Building upon the framework of *Corrective Unlearning* introduced by Goel et al. [2024], our work addresses a real-world scenario in which the defender has identified a small set of affected test points. In this work, we address the hardest of such settings where only *a single test point affected by the data poisoning attack is identifed*. Detecting such an affected test point is a natural prerequisite for initiating the unlearning process. This detection either occurs naturally during deployment due to the occurrence of abnormal predictions or via exhaustive in-house stress testing. Leveraging this poisoned test point, our approach comprises two primary tasks: first, identifying a critical set of manipulated training points responsible for the compromised prediction; and second, applying unlearning algorithms [Foster et al., 2024, Goel et al., 2023, Kurmanji et al., 2023] to remove the influence of these points from the model.

Within this framework, *influence functions* [Koh and Liang, 2017b] serve as a natural tool for attributing model predictions to specific training data points. However, recent studies [Bae et al., 2024, Grosse et al., 2023, Nguyen et al., 2024b] have indicated that state-of-the-art influence functions struggle to accurately identify the manipulated data points in deep learning models when used in a naive manner. Our experiments in Section 3 also corroborate this finding. To address this limitation, we introduce $\Delta - \text{Influence}$, a novel approach that enhances influence functions to reliably identify a sufficient critical set of training data points necessary for unlearning data poisoning without compromising model accuracy. Instead of directly calculating each training point's influence on a poisoned test point, $\Delta - \text{Influence}$ assesses the change in influence scores before and after perturbing the test point through label flipping and image transformation. As shown using ablation studies in Section 4, label flipping is essential for breaking the association between poisoned data and the compromised test point, while image transformations introduces randomness that reduces false positive rates by preserving the influence of benign data. These combined perturbations improve the detection of poisoned training samples.

To assess the effectiveness of $\Delta - \text{Influence}$ and the broader applicability of influence functions in this context, we apply our method to three prominent data poisoning attacks: BadNet [Gu et al., 2019], Witches' Brew [Geiping et al., 2021], and Smooth Trigger [Zeng et al., 2021]. We compare

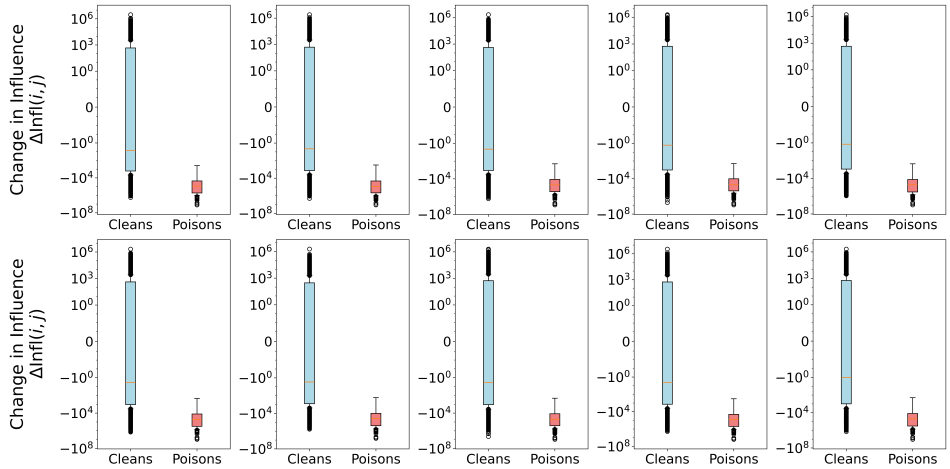

Figure 2: We show the Influence Score Change ($\Delta\text{Infl}(i, j)$) for 125 poisoned training points (orange) and 49,875 clean training points (light blue) on the Smooth Trigger attack with CIFAR100. Each plot shows the influence score change for a different transformation applied to the affected test image. Our result shows a consistent drop in influence scores for all poisoned examples after transformation, while clean examples exhibit no clear trend.

our approach against multiple defenses [Chen et al., 2018, Grosse et al., 2023, Tran et al., 2018, Zeng et al., 2021] that operate with similar or less information about the poisoning than $\Delta - \text{Influence}$. Each attack presents unique challenges for detection and mitigation, as evidenced by the varying performance of existing detection methods across different attacks. Additionally, we conduct experiments using several known unlearning algorithms to remove the data poisoning using the identified set. These experiments provides a comprehensive comparison of these unlearning algorithms highlight the strengths and weaknesses of each. For example, gradient ascent-based methods like SCRUB [Kurmanji et al., 2023] and weight deletion methods like SSD [Foster et al., 2024] can effectively unlearn poisoning when the detected set of training poisons is reasonably accurate. However, their resultant accuracy drops significantly if the detected set includes many falsely flagged clean examples. In contrast, methods like EU and CF [Goel et al., 2024] are surprisingly robust to false positives, delivering the best unlearning and accuracy. Overall, our experiments demonstrate that $\Delta - \text{Influence}$ consistently outperforms existing algorithms across all settings, offering a robust defense against sophisticated data poisoning attacks while preserving accuracy.

2 Using Influence functions to detect poisons

In this section, we present how influence functions can be leveraged to unlearn data poisoning attacks and introduce our primary algorithm, $\Delta - \text{Influence}$.

Consider a scenario where an adversary modifies a subset of training images belonging to a specific *victim* class by adding a subtle patch or trigger and altering their labels to a *target* class. These manipulated examples, referred to as *poisons*, are incorporated into the training dataset. Consequently, the trained model learns to misclassify any test image from the victim class containing the trigger as belonging to the target class, while maintaining normal performance on other test images.

Influence functions [Koh and Liang, 2017b] provide a mechanism to quantify the contribution of each training example to a particular prediction. By computing the influence of each training point on the prediction of the affected test point, we can identify the most influential training samples responsible for abnormal model behavior. Specifically, poisoned examples typically exert a significant influence on the affected test predictions, it makes it possible to distinguish the poisons through their influence scores. Thus, influence functions offer a natural approach to trace poisoned training data from misclassified test examples back to the responsible training instances.

However, our experiments in Section 3, along with several recent studies [Bae et al., 2024, Li et al., 2024b, Nguyen et al., 2024b], demonstrate that naively applying state-of-the-art influence functions [Grosse et al., 2023] fails to accurately identify poisoned points in deep neural networks.

This limitation necessitates the development of a more robust method to effectively utilize influence functions for detecting and unlearning data poisoning.

2.1 Our Algorithm: Δ-Influence

To address the shortcomings of the above naive approach, we introduce $\Delta - \text{Influence}$, a novel algorithm for influence-function based detection of poisoned training samples. The core idea of $\Delta - \text{Influence}$ is to monitor the changes in influence scores of training data points when the affected test point undergoes various transformations.

Notations. Let $z_{\text{tr}}^i := \left(x_{\text{tr}}^i, y_{\text{tr}}^i\right)$ denote a labeled training data point, and let $\theta^\star$ represent the trained model parameters optimized on the training dataset. For a given test point $z_{\text{te}} := (x_{\text{te}}, \hat{y}_{\text{te}})$ with predicted label $\hat{y}_{\text{te}}$, the influence function quantifying the impact of z_{tr}^i on the loss of z_{te} is:

$$\text{Infl}\left(\theta^\star, z_{\text{tr}}^i, z_{\text{te}}\right) = -\nabla_\theta \mathcal{L}\left(z_{\text{te}}, \theta^\star\right)^\top \mathbf{H}^{-1} \nabla_\theta \mathcal{L}\left(z_{\text{tr}}^i, \theta^\star\right), \tag{1}$$

where $\mathcal{L}(z, \theta^\star)$ is the loss evaluated at the point z with parameters $\theta^\star$ and $\mathbf{H}$ is the Hessian of the loss function with respect to θ at $\theta^\star$.

Monitoring Change in Influence. Our goal is to attribute the predicted label $\hat{y}_{\text{te}}$ of a poisoned test point z_{te} to a subset of training points $\mathcal{P} = \{z_{\text{tr}}^1, \ldots, z_{\text{tr}}^k\}$ responsible for the misclassification. To achieve this, $\Delta - \text{Influence}$ monitors the change in influence scores $\text{Infl}\left(\theta, z_{\text{tr}}^i, z_{\text{te}}\right)$ for each training data point z_{tr}^i when the test point z_{te} undergoes a set of transformations.

Formally, let g_j be a transformation applied to the test point $z_{\text{te}} = (x_{\text{te}}, y_{\text{te}})$, consisting of pairing the test image with a random label y'_{te} and applying standard data augmentations such as blurring, color jitter, rotating, and flipping to x_{te} (see Appendix B for a list of all transformations). In this work, we utilize common data augmentation techniques without designing any poison-specific transformations. We consider this simplicity of our approach and not needing to design special transformations to be a key strength of our work. Then, for each transformation g_j, compute the change in influence score as:

$$\begin{aligned}\Delta\text{Infl}(\theta, {z_{\text{tr}}}^i, g_j\left(z_{\text{te}}\right)) =& \text{Infl}\left(\theta, {z_{\text{tr}}}^i, g_j(z_{\text{te}})\right) \\ &- \text{Infl}\left(\theta, {z_{\text{tr}}}^i, z_{\text{te}}\right).\end{aligned} \tag{2}$$

For brevity, we denote this change as $\Delta\text{Infl}(i, j)$, where i and j index the training point and the transformation function, respectively.

Influence Collapse. Computing the Δ- Influence is motivated by the following two observations, which we refer to as *Influence Collapse*. Let z_{te} be the affected test point.

1. **Negative Change for Poisons:** For all manipulated training samples $z_{\text{tr}}^i \in \mathcal{P}$ and transformations g_j, the change in influence $\Delta\text{Infl}(i, j)$ is consistently negative.
2. **Minimal Change for Clean:** For all clean training samples $z_{\text{tr}}^k \notin \mathcal{P}$ and transformations g_j, the change in influence $\Delta\text{Infl}(k, j)$ is significantly smaller in magnitude and often positive in value, for most transformations.

This is illustrated in Figure 2, where $\Delta\text{Infl}(i, j)$ is consistently negative for poisoned samples across all transformations, whereas it often remains near zero (compared to that of poisons) or shows no clear trend for clean examples. However, Figure 2 shows that this is not consistently the case for all clean examples (with some values being considerably small), which brings us to the next component.

Boosting Using Multiple Transformations. The above discussion shows that the change in influence score $\Delta\text{Infl}(i, j)$ can be used as a score function for detecting whether z_{tr}^i is a manipulated training sample. However, this score function is a relatively weak classifier, especially for clean points, as seen in Figure 2. To overcome this problem, we use classical ideas from bagging and apply multiple transformations $g_1, \ldots, g_{n_b}$ to obtain a series of weak score functions. Specifically, we use n_b transformations to obtain n_b weak score functions.

Upon obtaining the scores $\Delta\text{Infl}(i, j)$ for all transformations $j = 1$ to $j = n_b$, we combine the scores using a count-based decision rule. Specifically, if a sufficiently large number of transformations lead

to a negative change in influence score, we flag the training data as manipulated. The key hypothesis we leverage here is that for most clean points, a few transformations will always result in a positive change in influence $\Delta\text{Infl}(i, j)$.

Unlearning identified points. Once the set of poisoned training points $\mathcal{P}$ is identified using $\Delta - \text{Influence}$, the next step is to unlearn these points to mitigate the data poisoning attack. We employ several unlearning algorithms [Foster et al., 2024, Goel et al., 2023, Golatkar et al., 2020, Kurmanji et al., 2023] to remove the influence of $\mathcal{P}$ from the trained model $\theta^\star$. In practice, the choice of unlearning algorithm may depend on factors such as computational efficiency, scalability, and the specific characteristics of the poisoning attack. In this work, we look at several popular algorithms including retraining from scratch (denoted as EU [Goel et al., 2023]), CF [Goel et al., 2023], SSD [Foster et al., 2024], SCRUB [Golatkar et al., 2020], and BadT [Kurmanji et al., 2023].

2.2 Full Algorithm

To summarise, the full pipeline of detection and unlearning in $\Delta - \text{Influence}$ proceeds as follows:

1. Initialization Begin with trained model $\theta^\star$, a poisoned test point z_{te}, and the entire training dataset $\mathcal{D} = \{z_{\text{tr}}^i\}_{i=1}^N$.
2. Transformations Apply a diverse set of transformations $\mathcal{G} = \{g_j\}_{j=1}^{n_b}$ to the poisoned test point z_{te} to obtain multiple $z'_{\text{te}} = g_j(z_{\text{te}})$.
3. Influence Score For each training data point $z_{\text{tr}}^i \in \mathcal{D}$ and each transformation $g_j \in \mathcal{G}$, compute the change in influence score $\Delta\text{Infl}(i, j)$ as defined in Equation (2).
4. Boosting and Detection For each training data point z_{tr}^i, aggregate the influence score changes across all transformations. If the number of negative changes exceeds a predefined threshold τ, flag z_{tr}^i as a poisoned sample.
5. Unlearning Once the set of poisoned training points $\mathcal{P}$ is identified, apply unlearning algorithms to remove their influence from the trained model $\theta^\star$.

In the next section, we use the above algorithm for experiments on several data poisons, datasets, and unlearning algorithms and compare them with existing approaches.

3 Experiments

We now showcase the empirical performance of our algorithm in comparison to multiple baselines.

3.1 Experimental Setup

Attacks. To ensure broad coverage and robustness, we evaluate our $\Delta - \text{Influence}$ algorithm against three distinct types of data poisoning attacks:

1. Patch Trigger (BadNet) [Gu et al., 2019]: Also studied in Goel et al. [2024], this attack involves adding a small, subtle patch to the corner of selected training images and altering their labels to a designated target class. The presence of the patch causes the trained model to misclassify any test image containing the patch into the target class while maintaining normal performance on other inputs.
2. Frequency Trigger [Zeng et al., 2021]: In this approach, a trained, imperceptible pattern is embedded both the spatial and frequency domains, thereby encompassing the whole image unlike BadNet. As shown in Alex et al. [2024]. these patterns are difficult to detect by both human observers and automated detection methods, making the poisoned samples challenging to identify and remove.
3. Clean Label Attack (Witches' Brew) [Geiping et al., 2021]: Unlike BadNets and Frequency Trigger, this attack adds an imperceptible pattern to images without altering their labels. The poisoned samples appear benign since their labels are consistent with their content, yet they cause the model to learn incorrect associations, leading to misclassifications during inference. As shown in Pawelczyk et al. [2024], these patterns are difficult to unlearn using unlearning algorithms.

Model and Datasets. We utilize the CIFAR10 and CIFAR100 datasets [Krizhevsky] and a ResNet18 model [He et al., 2015], following the standard benchmarks and models used in the state-of-the-art

machine unlearning setup [Pawelczyk et al., 2024]. For CIFAR10, we poison 500 training images (1% of the dataset), while for CIFAR100, we poison 125 training images (0.25% of the dataset) for all attack types except BadNet, which requires a higher size of 350 samples to be effective. The victim class and attack class (when different) are selected randomly. Detection methods are tuned on a small validation set using cross-validation techniques. Hyperparameters such as threshold values and clustering parameters are optimized based on validation performance metrics to achieve the best balance between detection accuracy and false positive rates. Detailed hyperparameter settings are provided in the Appendix B to ensure reproducibility. The code will be made publicly available.

Compared Methods. We compare the detection performance of existing popular methods in the data poisoning literature by adapting them to our setting. Additionally, we include the state-of-the-art influence function method EK-FAC [Grosse et al., 2023] as a baseline. Our $\Delta - \text{Influence}$ method is built upon EK-FAC to ensure fairness in the influence computation itself. This allows us to provide a fair comparison of our algorithm against naive influence function.

1. Activation Clustering-Based Detection [Chen et al., 2018] identifies backdoored samples by clustering the activations of the last hidden layer for each class. If a class's activations can be effectively clustered into two distinct groups, the smaller cluster is deemed to contain poisoned samples and is subsequently removed for retraining.
2. Spectral Signature-Based Detection [Tran et al., 2018] employs singular value decomposition on the activations of the last hidden layer per class. Samples with high values in the first singular dimension are flagged as poisoned and removed based on a predefined hyperparameter threshold.
3. Frequency-Based Detection [Zeng et al., 2021] performs frequency analysis by building a classifier on the discrete cosine transforms of synthetic images containing hardcoded backdoor-like features. It identifies poisoned examples by detecting these frequency-based patterns.
4. EK-FAC [Grosse et al., 2023] serves as our baseline method for using influence functions in poison detection. It calculates influence scores for every training sample based on one known affected test sample. Samples with average influence scores exceeding a predefined threshold are removed.

Metrics. We evaluate our algorithm using four key metrics. All metrics are averaged over three runs with different random seeds.

1. True Positive Rate: Fraction of identified poisoned samples out of the total poisoned samples in train set.
$$\frac{\text{Number of correctly flagged poisoned samples}}{\text{Total number of poisoned samples}} \times 100\%$$
2. Precision: Proportion of correctly identified poisoned samples among all samples flagged as poisoned. It reflects the trade-off between detection accuracy and model utility.
$$\frac{\text{Number of correctly flagged poisoned samples}}{\text{Total number of samples flagged as poisoned}} \times 100\%$$
3. Poison Success Rate: Fraction of poisoned test samples that are misclassified into the target (incorrect) class. For the Witches' Brew attack, it indicates whether the test point remains poisoned.
$$\frac{\text{Number of poisoned samples classified as target}}{\text{Total number of poisoned samples}} \times 100\%$$
4. Test Accuracy: The Model's performance on unpoisoned test samples, measuring drop in model utility.
$$\frac{\text{Number of correct predictions on test set}}{\text{Total number of test samples}} \times 100\%$$

3.2 Main Results

We present our experimental findings across the above metrics and compare the performance of $\Delta - \text{Influence}$ against baseline methods. Specifically, we report the fraction of detected poisoned samples and the true positive rate in Table 1, and the overall poison success rate along with test accuracy in Figure 3.

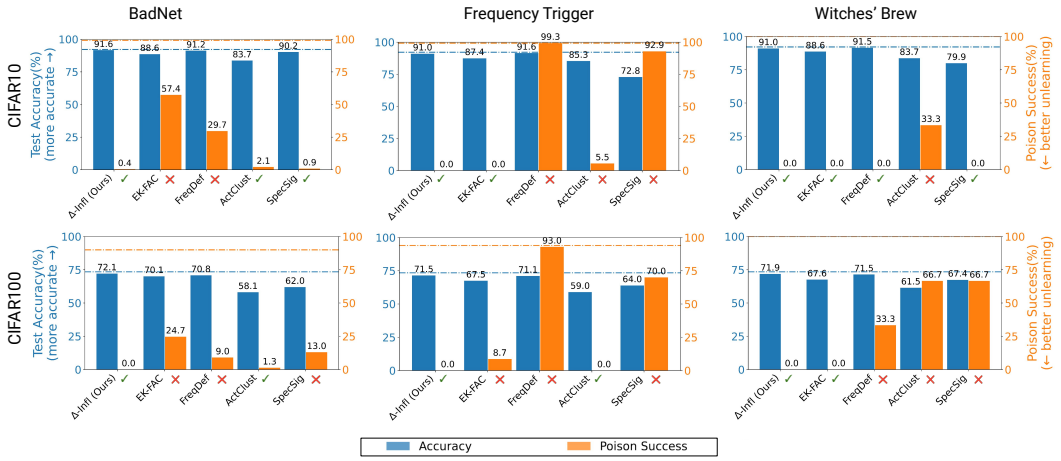

Figure 3: Poison Success Rate and Test Accuracy. This table shows both poison unlearning effectiveness and model utility. A method is considered successful if the poison success rate is below 5%, marked by ✓, with unsuccessful methods marked by ×. Δ-Influence is successful in 6/6 cases, while the closest competitors succeed in only 3/6. Additionally, Δ-Influence nearly perfectly preserves test accuracy. Figure structure from [Pawelczyk et al., 2024].

Method	**Metric**	**CIFAR10** BadNet	**CIFAR10** Frequency Trigger	**CIFAR10** Witches' Brew	**CIFAR100** BadNet	**CIFAR100** Frequency Trigger	**CIFAR100** Witches' Brew
SpecSig	Precision	3.6%	1.3%	1.4%	1.3%	0.5%	0.3%
	TPR	88.3%	88.3%	96.8%	82.6%	78.4%	35.2%
ActClust	Precision	2.2%	2.2%	2.1%	1.6%	0.6%	0.3%
	TPR	94.9%	99.1%	93.4%	96.3%	100%	55.2%
FreqDef	Precision	8.0%	0.4%	10.2%	5.3%	0.1%	1.8%
	TPR	72.3%	3.2%	93.6%	85.7%	2.4%	78.4%
EK-FAC	Precision	2.8%	2.9%	0.8%	3.2%	0.9%	0.4%
	TPR	67.1%	100%	17.4%	70.0%	96.8%	47.2%
Δ-Infl (Ours)	Precision	17.6%	13.3%	3.3%	37.3%	2.9%	2.1%
	TPR	99.1%	100%	19.4%	96.9%	100%	62.4%

Table 1: Comparison of Precision & TPR across methods and dataset for detecting poisoned samples. Green indicates successful unlearning (poisoning success rate $\leq 5\%$, while red indicates unsuccessful unlearning (see Figure 3 for exact poisoning success rates). We evaluate the precision and true positive rate (TPR) of detecting poisoned training samples. SpecSig [Tran et al., 2018], ActClust [Chen et al., 2018], and EK-FAC [Grosse et al., 2023] yield low precision, flagging many clean samples as poisoned. FreqDef [Zeng et al., 2021] and $\Delta - \text{Influence}$ (Ours) better preserve clean data, though FreqDef shows a significantly lower TPR, missing many true poisoned samples.

Performance of Δ-Influence. As illustrated in Figure 3, Δ-Influence consistently achieves a poison success rate below 2% across all three types of poisoning attacks and both datasets. This success rate is marked by a ✓, while unsuccessful detections are marked by a ×. In contrast, the next best methods, Activation Clustering (ActClust) and EK-FAC, succeed in only 3 out of 6 cases, as highlighted in Table 1. This showcases the substantial improvement in performance of $\Delta - \text{Influence}$.

Among the baseline methods, EK-FAC outperforms ActClust by minimizing the drop in test accuracy, which is indicated by a higher precision in Table 1. Furthermore, Δ-Influence consistently achieves the highest precision, offering better performance with minimal accuracy loss compared to the other methods. Additional experiments detailed in Section 4.1 demonstrate that both label and input augmentations are necessary for $\Delta - \text{Influence}$.

Variance across Poisons. Our analysis shows that the BadNet poison can be effectively removed without identifying all poisoned samples, reaffirming that it is realatively easy to eliminate. Based on these results, we advocate that the corrective unlearning literature should benchmark proposed algorithms on the more challenging frequency-based poisons [Zeng et al., 2021], which require detecting nearly all poisoned samples and are notably harder to remove with a partial subset. This was also identified to be difficult in previous work [Alex et al., 2024].

Surprisingly, in the case of the Witches' Brew attack on CIFAR-10, our $\Delta - \text{Influence}$ method often identifies fewer but a sufficient number of true poisoned samples compared to other methods. We

Method	Metric	CIFAR10			CIFAR100		
		BadNet	Frequency Trigger	Witches' Brew	BadNet	Frequency Trigger	Witches' Brew
Ours (Label-Only)	Precision	4.0%	6.3%	1.2%	3.1%	1.1%	0.8%
	TPR	97.5%	100%	24.2%	99.1%	100%	73.6%
Ours (Img-Only)	Precision	14.4%	31.3%	2.5%	7.6%	0.0%	1.1%
	TPR	68.9%	8%	3.2%	50.6%	0.0%	20.8%
Ours (Both)	Precision	17.6%	13.3%	3.3%	37.3%	2.9%	2.1%
	TPR	99.1%	100%	19.4%	96.9%	100%	62.4%

Table 2: Comparison of Precision and TPR across Label-Only, Image-Only and combined transformation of affected image. Green indicates successful unlearning (poisoning success rate $< 5\%$), while red indicates unsuccessful unlearning (See Appendix Figure 7 for exact poison success rate). Label-only augmentations are highly effective in detecting poisoned samples, whereas image-only augmentations perform poorly. Conversely, image-only augmentations significantly reduce the false positive rate, preserving more clean data and improving detection precision.

attribute this to the unique behavior of this particular poison. $\Delta - \text{Influence}$ effectively identifies the samples most responsible for the misclassification, and in Witches' Brew, only a few samples are truly effective for poisoning. Additional experiments in Section 4.3 show that removing the complement of detected poisons does not allow the model to recover, despite the complement set being similar in size or larger.

Conclusion. Overall, $\Delta - \text{Influence}$ offers an effective mechanism for unlearning data poisonining attacks without significantly impacting model performance.

4 Unpacking Key Factors in Δ-Infleunce

In this section, we present a series of additional analyses designed to improve the understanding of our proposed $\Delta - \text{Influence}$ method. Specifically, we explore: (i) individual contributions of image and label perturbations, (ii) effectiveness of various unlearning algorithms, and (iii) a counterfactual analysis to determine whether the detected samples are solely responsible for enabling poisoning in the Witches' Brew attack.

4.1 Perturbing Only Images or Labels

Setup. To distinguish the contributions of image and label perturbations in our $\Delta - \text{Influence}$ method, we conduct an ablation study by evaluating the two key components separately:

1. Modify Image ($\Delta - \text{Influence}$ (Img-Only)): In this baseline, we exclusively modify the test images without altering their labels. This allows us to isolate the impact of image transformations on the model's ability to detect poisoned data.

2. Modify Label ($\Delta - \text{Influence}$ (Label-Only)): Conversely, in this baseline, we only modify the test point's labels while keeping the images unchanged. This setup helps evaluate the effect of label manipulation on detecting the influence of poisoned training points.

Both ablations are benchmarked across the same datasets and poisoning attacks, utilising identical metrics to ensure consistency in evaluation. The goal is to understand the individual and combined effects of image and label perturbations on the detection performance of $\Delta - \text{Influence}$.

Results. As depicted in Table 2, our ablation study reveals that label-only augmentations achieve high TPR across all poisoning types and datasets, effectively identifying almost all poisoned samples. However, this leads to low precision, resulting in the unnecessary removal of a significant number of clean samples. On the other hand, image-only augmentations exhibit poor TPR, failing at the core task but also rejects lesser number of clean samples (higher precision). In contrast, the combined approach ($\Delta - \text{Influence}$ (Ours)) leverages both label and image perturbations to achieve a balanced performance and successfully detects a high number of poisoned samples (high TPR) while rejecting lesser clean samples (high precision).

Conclusion. Our ablation study underscores the necessity of incorporating both label and image augmentations in the $\Delta - \text{Influence}$ method. Label perturbations are pivotal for enhancing detection accuracy, while image augmentations play a critical role in minimizing false positives.

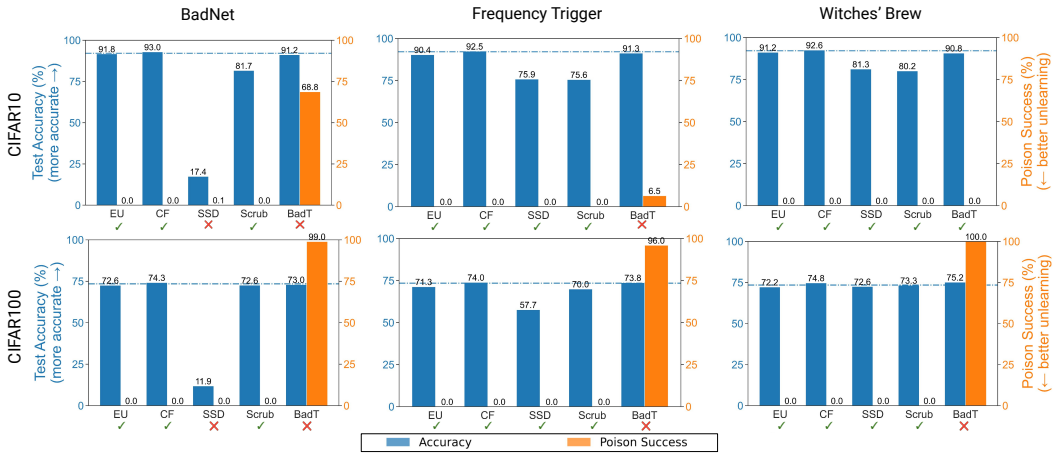

Figure 4: Poison Success Rate and Test Accuracy for Unlearning Methods Applied on Samples Identified by $\Delta - \text{Influence}$. Catastrophic Forgetting (CF) and Exact Unlearning (EU) from Goel et al. [2023] perform best, effectively unlearning poisoned samples while maintaining test accuracy. In contrast, SSD [Foster et al., 2024] and SCRUB [Kurmanji et al., 2023] struggle with false negatives, leading to significant accuracy drops, while BadT [Chundawat et al., 2023] fails to unlearn effectively. We recommend EU or CF as strong baselines and highlight the need for future methods to improve robustness against false positives.

4.2 Counterfactual Analysis: Do Detected Samples Account for Poisoning in Witches' Brew?

Δ-Influence Set	TPR(↑)	Poison Success Rate (↓)	Test Accuracy (↑)
CIFAR10			
Original	19.4%	0%	91.03%
Complement	80.6%	100%	92.23%
CIFAR100			
Original	62.4%	0%	71.92%
Complement Set	37.6%	100%	72.79%

Table 3: **Does the Detected Set Truly Influence the Poison?** For Witches' Brew, we test the "Original" set, representing the poisoned samples identified by $\Delta - \text{Influence}$, and the "Complement" set, which includes all other poisoned samples not detected. The absence of a drop in poison success rate when removing the complement set suggests that the detected set fully captures the poisoning effect. Conversely, removing the detected set completely eliminates the poisoning effect.

Setup. We perform a counterfactual analysis by comparing the original detected set of poisoned samples in the Witches' Brew attack to its complement set (i.e., all poisoned samples except those detected by $\Delta - \text{Influence}$). This comparison aims to assess whether the detected set exclusively accounts for the poisoning effect.

Results. As presented in Table 3, the removal of the "Original" detected set (19.4% TPR for CIFAR10 and 62.4% TPR for CIFAR100) results in 0% poison success rate, effectively unlearning the poisoning. In stark contrast, removing the "Complement" set (80.6% TPR for CIFAR10 and 37.6% TPR for CIFAR100) maintains a poison success rate of 100%, indicating that the undetected samples do not sufficiently contribute to the poisoning. The complement set achieves higher test accuracy simply because it only contains unaffected samples without false positives.

Conclusion. These results demonstrate that our detected subset accounts for nearly all the poisoning effects in Witches' Brew, highlighting the unusual nature of the poison and the precision of our $\Delta - \text{Influence}$ algorithm.

5 Investigating Unlearning Methods

This section examines the impact of unlearning methods by evaluating the effectiveness of $\Delta - \text{Influence}$ across various unlearning techniques. Additionally, we investigate how employing a

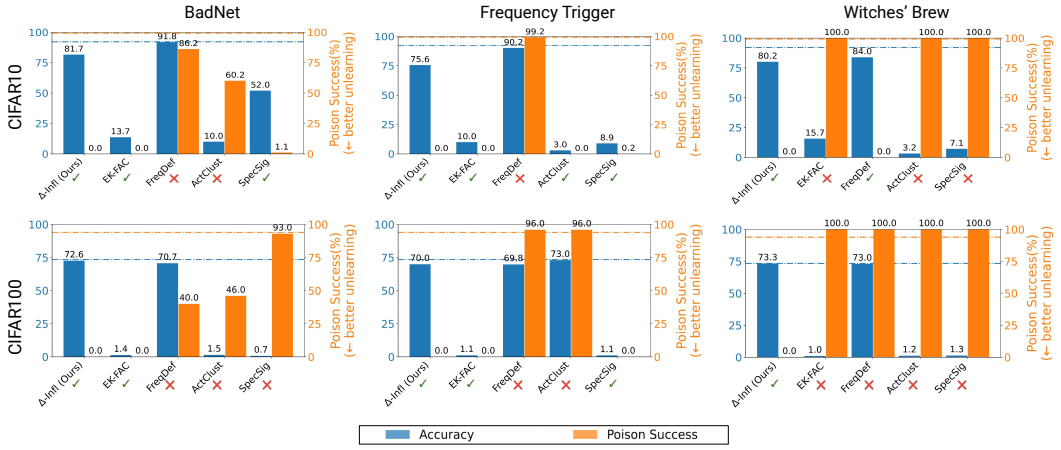

Figure 5: Poison Success Rate and Test Accuracy. with SCRUB Unlearning algorithm. This table shows both poison unlearning effectiveness and model utility. A method is considered successful if the poison success rate is below 5%, marked by ✓, with unsuccessful methods marked by ×. Δ-Influence is successful in 6/6 cases, while the rest fail by not be distinguishable from a randomly initialized model. In contrast, Δ-Influence has only minor drops in test accuracy. Figure structure from [Pawelczyk et al., 2024].

different unlearning algorithm could influences both the success rate of the poisoning attacks and the performance outcomes when applied to various detection methods.

5.1 Which Unlearning Methods Work?

Setup. To evaluate the effectiveness of various unlearning algorithms when paired with our $\Delta - \text{Influence}$ method, we fix the influence function to $\Delta - \text{Influence}$ and vary the unlearning functions. We benchmark several corrective unlearning methods, including EU [Goel et al., 2023], CF [Goel et al., 2023], SSD [Foster et al., 2024], SCRUB [Golatkar et al., 2020], and BadT [Kurmanji et al., 2023]. All methods are implemented using the codebase and training protocols from Goel et al. [2024]. Further implementation details, including parameter settings and computational resources, are provided in the Appendix.

Results. As illustrated in Figure 4, our evaluation reveals that CF performs comparably to EU, achieving similar poison removal success rates while offering significant computational gains by avoiding full retraining. CF remains robust against false positives, maintaining high test accuracy. EU effectively removes poisoned samples with no significant drop in test accuracy, albeit at a higher computational cost due to retraining. In contrast, while gradient-ascent-based methods like SCRUB and weight deletion approaches like SSD successfully unlearn poisons, they do so at the expense of model utility due to their susceptibility to false positives. Finally, BadT fails to unlearn poisons effectively.

Conclusion. We recommend EU or CF [Goel et al., 2023] as competitive baselines for corrective unlearning setting using influence functions, and also highlight the importance of robustness towards false positives.

5.2 Does Δ-Influence Perform the Best Across Unlearning Algorithms?

Setup. The probe was conducted across various detection methods; however, instead of employing the exact unlearning algorithm, we use a popular alternative algorithm called SCRUB which involves gradient ascent. We similarly measure the performance as well as the success rate of the poison removal were evaluated. Note the TPR rate and precision do not change.

Results. The evaluation results in Figure 5 shows that $\Delta - \text{Influence}$ outperforms other methods, unlearning poisons in all six cases with minimal performance loss. In contrast, EK-FAC, ActClust, and SpecSig performed randomly, achieving unlearning primarily because even a randomly initialized model would not retain poisoning. Performance drops were primarily due to SCRUB's sensitivity to false positives from its gradient ascent step. FreqDef avoided randomness but failed to unlearn

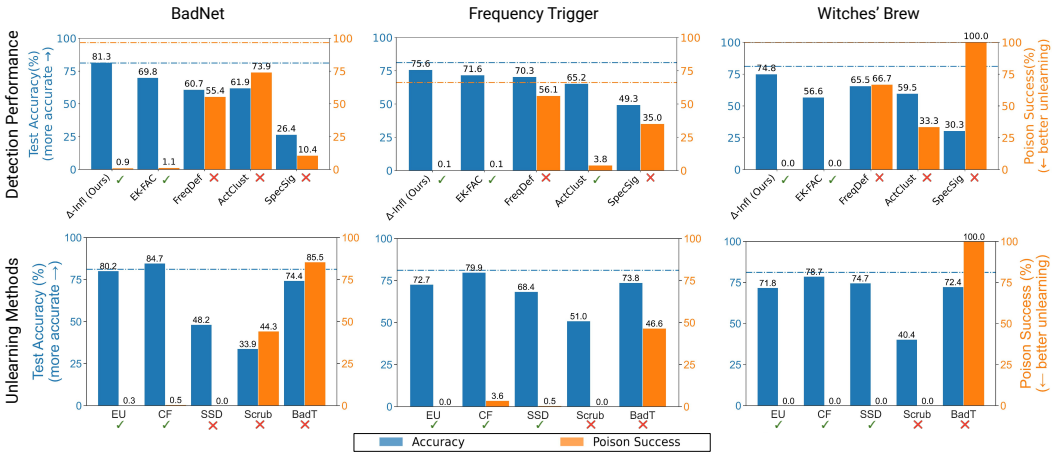

Figure 6: **Scaling to Imagenette.** In the top row, results on Imagenette are consistent with previous findings: $\Delta - \text{Influence}$ effectively unlearns all three types of poisons while preserving high test accuracy. In contrast, other detection methods often fail to unlearn or do so at the expense of test accuracy. In the bottom row, EU and CF consistently perform well, with CF [Goel et al., 2023] achieving the smallest accuracy drop and only a minor increase in poison success rate.

poisons in all cases. Notably, $\Delta - \text{Influence}$ minimized false positives, maintaining consistent and reliable outcomes.

Conclusions. $\Delta - \text{Influence}$ proves to be remarkably robust even across unlearning methods which are highly sensitive to false positives. It achieves a 6/6 poison removal rate while incurring only minor performance losses due to false positives.

6 Scaling Findings to ImageNette

Setup. To evaluate the scalability and consistency of our $\Delta - \text{Influence}$ algorithm on a more complex and larger dataset, we conduct experiments on Imagenette. The setup is consistent with the experiments in Section 3 with specific adjustments to accommodate Imagenette's larger image sizes and increased complexity. Specifically, we increase the patch size for BadNet poisoning, use a larger trigger pattern for frequency-based poisoning, and poison a greater fraction of training images (10%). Additionally, for the Witches' Brew method, we relax the perturbation constraint, setting $\epsilon = 32$ instead of $\epsilon = 16$.

Results. Replicating our prior experiments on Imagenette, Figure 6 illustrates that $\Delta - \text{Influence}$ continues to achieve the most effective poison unlearning across all attack types, maintaining minimal accuracy loss. Notably, the EK-FAC baseline also successfully unlearns all poisons but incurs a higher false positive rate, leading to significant drops in test accuracy due to the unnecessary removal of clean samples. Additionally, when applying various unlearning algorithms to the samples identified by $\Delta - \text{Influence}$, both CF and EU perform consistently well with CF achieving notably higher accuracy during poison unlearning compared to EU.

Conclusion. Scaling to larger datasets preserves all prior conclusions, underscoring the robustness of our results and the effectiveness of $\Delta - \text{Influence}$ and CF-based unlearning methods.

6.1 Scaling to Larger Set of Identified Poisoned Test Samples

Setup. For attack methods such as Witches' Brew, only a single affected test point is identified. However, in cases where multiple test points can be identified, such as with BadNet Patch and Smooth Trigger attacks, we explore ways to enhance performance using two influence-based methods: $\Delta - \text{Influence}$ and EK-FAC, on the ImageNette dataset. Specifically, we select five test points to identify corresponding input points and determine their intersection as the poisoned data across both methods. This is done similarly to the $\Delta - \text{Influence}$ algorithm by retaining points with influence higher than the tolerance threshold, hence EK-FAC is additionally labeled (boosted).

Influence Methods	Precision(↑)	TPR(↑)	Poison Success Rate (↓)	Test Accuracy (↑)
1 identified test point				
EK-FAC	22.1%	99.1%	0.3%	68.7%
Δ-Influence	49.0%	100%	0.8%	79.7%
5 identified test points				
EK-FAC	25.9%	98.8%	0.5%	73.3%
EK-FAC(boosted)	34.2%	98.5%	0.8%	75.4%
Δ-Influence	66.7%	100%	0.5%	80.0%
10 identified test points				
EK-FAC	26.6%	98.8%	0.5%	75.8%
EK-FAC(boosted)	48.9%	97.2%	1.6%	77.8%
Δ-Influence	67.2%	100%	0.8%	79.9%

Table 4: **ImageNette BadNet.** On the ImageNette dataset, increasing the number of identified test points significantly improves the true positive rate (TPR). This enhancement leads to a notable reduction in false positives, thereby achieving higher overall test accuracy.

Influence Methods	Precision(↑)	TPR(↑)	Poison Success Rate (↓)	Test Accuracy (↑)
1 identified test point				
EK-FAC	10.5%	99.3%	0%	72.4%
Δ-Influence	25.8%	99.3%	0%	75.4%
5 identified test points				
EK-FAC	12.8%	99.0%	0%	74.4%
EK-FAC(boosted)	21.8%	99.0%	0.3%	74.0%
Δ-Influence	27.5%	99.3%	0.3%	76.6%
10 identified test points				
EK-FAC	12.9%	99.3%	0%	74.1%
EK-FAC(boosted)	24.2%	99.3%	0.3%	73.6%
Δ-Influence	28.7%	99.3%	0.3%	75.0%

Table 5: **ImageNette Smooth Trigger.** On the ImageNette dataset, increasing the number of identified test points significantly improves the true positive rate (TPR). This enhancement leads to a notable reduction in false positives, thereby achieving higher overall test accuracy.

Results. We showcase performance in Table 4 for BadNet poison and Table 5 for frequency trigger poison respectively. We observe a consistent trend: as the set of identified poisons increases, the true positive rate (TPR) improves significantly, leading to a substantial reduction in false positives and ultimately higher test accuracy. Overall, identifying multiple poisoned test points enables more precise detection of poisons in the training set when using $\Delta - \mathrm{Influence}$-like aggregation algorithms across test poisoned points.

7 Conclusion

In this study, we address a critical issue in corrective machine unlearning: identifying key training samples whose removal can unlearn a data poisoning attack. We address a practical scenario where only a limited number of affected test points are known—potentially discovered post-deployment or through internal testing. To this end, we introduce $\Delta - \mathrm{Influence}$, a novel approach that uses influence functions to trace abnormal model behavior back to the responsible poisoned training data, requiring as little as one affected test example. By retraining without these identified points, Δ-Influence successfully unlearns multiple poisoning attacks across diverse datasets. We evaluate our method against five state-of-the-art detection algorithms and apply five well known unlearning algorithms to the identified training set. Our results demonstrate that Δ-Influence consistently outperforms existing approaches in all tested scenarios. Our findings highlight the potential of influence functions as a foundation for unlearning data poisoning attacks. Additionally, our ablation study sheds light on the strengths and limitations of various poisoning attacks and unlearning algorithms, offering insights that could inform the development of more effective unlearning techniques and robust poisoning attacks for rigorous testing.

Acknowledgements

The authors would like to thank (in alphabetic order): Shashwat Goel, Shyamgopal Karthik, Elisa Nguyen, Shiven Sinha, Shashwat Singh, Matthias Tangemann, Vishaal Udandarao for their helpful feedback. WL, JL, and CSW acknowledges support from the Supervised Program for Alignment Research (SPAR) research program. We also acknowledge the Center for AI Safety (CAIS) for their support in providing the computational resources necessary for this study.

References

Neel Alex, Shoaib Ahmed Siddiqui, Amartya Sanyal, and David Krueger. Protecting against simultaneous data poisoning attacks. *arXiv:2408.13221*, 2024. 5, 7, 17

Juhan Bae, Wu Lin, Jonathan Lorraine, and Roger Grosse. Training data attribution via approximate unrolled differentation. *arXiv:2405.12186*, 2024. 2, 3

Samyadeep Basu, Phil Pope, and Soheil Feizi. Influence functions in deep learning are fragile. In *International Conference on Learning Representations (ICLR)*, 2021. 16

Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine Unlearning. *IEEE Symposium on Security and Privacy (IEEE S&P)*, 2021. 16

Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *IEEE Symposium on Security and Privacy (IEEE S&P)*, 2015. 16

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 (IEEE S&P)*, 2024. 16

Raja Chatila, Virginia Dignum, Michael Fisher, Fosca Giannotti, Katharina Morik, Stuart Russell, and Karen Yeung. Trustworthy AI. In *Reflections on Artificial Intelligence for Humanity*, 2021. 1

Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian Molloy, and Biplav Srivastava. Detecting backdoor attacks on deep neural networks by activation clustering. *arXiv:1811.03728*, 2018. 3, 6, 7

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 Conferences on Artificial Intelligence (IJCAI)*, 2019. 16

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*, 2023. 9

R Dennis Cook and Sanford Weisberg. Characterizations of an empirical influence function for detecting influential cases in regression. *Technometrics*, 1980. 16

Thomas Davenport and Ravi Kalakota. The potential for artificial intelligence in healthcare. *Future Healthc J*, 2019. 1

Yinpeng Dong, Xiao Yang, Zhijie Deng, Tianyu Pang, Zihao Xiao, Hang Su, and Jun Zhu. Black-Box Detection of Backdoor Attacks With Limited Information and Data. In *International Conference on Computer Visions (ICCV)*, 2021. 16

Jiaxin Fan, Qi Yan, Mohan Li, Guanqun Qu, and Yang Xiao. A Survey on Data Poisoning Attacks and Defenses. In *IEEE International Conference on Data Science in Cyberspace (DSC)*, 2022. 1

Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 16

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*, 2024. 2, 3, 5, 9, 10

Jonas Geiping, Liam H Fowl, W. Ronny Huang, Wojciech Czaja, Gavin Taylor, Michael Moeller, and Tom Goldstein. Witches' brew: Industrial scale data poisoning via gradient matching. In *International Conference on Learning Representations*, 2021. 2, 5, 16, 17

Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In *International conference on machine learning*, 2019. 16

Antonio Ginart, Melody Y. Guan, Gregory Valiant, and James Zou. Making AI forget you: Data deletion in machine learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 16

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. 2, 5, 9, 10, 11

Shashwat Goel, Ameya Prabhu, Philip Torr, Ponnurangam Kumaraguru, and Amartya Sanyal. Corrective machine unlearning. *Transactions on Machine Learning Research*, 2024. 1, 2, 3, 5, 10, 16, 17

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*, 2020. 5, 10

Roger Grosse, Juhan Bae, Cem Anil, Nelson Elhage, Alex Tamkin, Amirhossein Tajdini, Benoit Steiner, Dustin Li, Esin Durmus, Ethan Perez, Evan Hubinger, Kamilė Lukošiūtė, Karina Nguyen, Nicholas Joseph, Sam McCandlish, Jared Kaplan, and Samuel R. Bowman. Studying Large Language Model Generalization with Influence Functions. *arXiv:2308.03296*, 2023. 1, 2, 3, 6, 7, 16

Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Evaluating backdooring attacks on deep neural networks. *arXiv:1708.06733*, 2019. 2, 5, 16

Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens van der Maaten. Certified data removal from machine learning models. In *International Conference on Machine Learning (ICML)*, 2020. 16

Wenbo Guo, Lun Wang, Xinyu Xing, Min Du, and Dawn Song. TABOR: A Highly Accurate Approach to Inspecting and Restoring Trojan Backdoors in AI Systems, 2019. 16

Varun Gupta, Christopher Jung, Seth Neel, Aaron Roth, Saeed Sharifi-Malvajerdi, and Chris Waites. Adaptive machine unlearning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 16

Zayd Hammoudeh and Daniel Lowd. Training data influence analysis and estimation: A survey. *Machine Learning*, 2024. 16

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. 2015. 5

Jian Huang, Junyi Chai, and Stella Cho. Deep learning in finance and banking: A literature review and classification. *Frontiers of Business Research in China*, 2020. 1

Andrew Ilyas, Sung Min Park, Logan Engstrom, Guillaume Leclerc, and Aleksander Madry. Datamodels: Predicting predictions from training data. *arXiv preprint arXiv:2202.00622*, 2022. 16

Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nick Hynes, Nezihe Merve Gürel, Bo Li, Ce Zhang, Dawn Song, and Costas J Spanos. Towards efficient data valuation based on the shapley value. In *The 22nd International Conference on Artificial Intelligence and Statistics*, 2019. 16

SungYub Kim, Kyungsu Kim, and Eunho Yang. Gex: A flexible method for approximating influence via geometric ensemble. *Advances in Neural Information Processing Systems*, 2024. 16

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, 2017a. 16

Pang Wei Koh and Percy Liang. Understanding Black-box Predictions via Influence Functions. In *Proceedings of the 34th International Conference on Machine Learning*, 2017b. 2, 3, 16

Pang Wei W Koh, Kai-Siang Ang, Hubert Teo, and Percy S Liang. On the accuracy of influence functions for measuring group effects. *Advances in neural information processing systems*, 2019. 16

Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. 5

Meghdad Kurmanji, Peter Triantafillou, and Eleni Triantafillou. Towards unbounded machine unlearning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 2, 3, 5, 9, 10

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:2403.03218*, 2024a. 16

Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Anti-Backdoor Learning: Training Clean Models on Poisoned Data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021a. 17

Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Neural Attention Distillation: Erasing Backdoor Triggers from Deep Neural Networks. In *International Conference on Learning Representations*, 2021b. 16

Zhe Li, Wei Zhao, Yige Li, and Jun Sun. Do influence functions work on large language models? *arXiv:2409.19998*, 2024b. 3

Seth Neel, Aaron Roth, and Saeed Sharifi-Malvajerdi. Descent-to-delete: Gradient-based methods for machine unlearning. In *Conference on Learning Theory (COLT)*, 2021. 16

Elisa Nguyen, Johannes Bertram, Evgenii Kortukov, Jean Y Song, and Seong Joon Oh. Towards user-focused research in training data attribution for human-centered explainable ai. *arXiv preprint arXiv:2409.16978*, 2024a. 1

Elisa Nguyen, Minjoon Seo, and Seong Joon Oh. A bayesian approach to analysing training data attribution in deep learning. *Advances in Neural Information Processing Systems*, 2024b. 2, 3

Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. Trak: Attributing model behavior at scale. *arXiv:2303.14186*, 2023. 16

Martin Pawelczyk, Jimmy Z. Di, Yiwei Lu, Gautam Kamath, Ayush Sekhari, and Seth Neel. Machine Unlearning Fails to Remove Data Poisoning Attacks. *arXiv:2406.17216*, 2024. 1, 2, 5, 6, 7, 10, 16, 18

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. *Advances in Neural Information Processing Systems*, 2020. 16

Stefan Schoepf, Jack Foster, and Alexandra Brintrup. Potion: Towards poison unlearning. *arXiv:2406.09173*, 2024. 16

Sanjay Seetharaman, Shubham Malaviya, Rosni Vasu, Manish Shukla, and Sachin Lodha. Influence Based Defense Against Data Poisoning Attacks in Online Learning. In *International Conference on COMmunication Systems & NETworkS (COMSNETS)*, 2022. 16

Ayush Sekhari, Jayadev Acharya, Gautam Kamath, and Ananda Theertha Suresh. Remember what you want to forget: Algorithms for machine unlearning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 16

Mohsen Soori, Behrooz Arezoo, and Roza Dastres. Artificial intelligence, machine learning and deep learning in advanced robotics, a review. *Cognitive Robotics*, 2023. 1

Jacob Steinhardt, Pang Wei W Koh, and Percy S Liang. Certified Defenses for Data Poisoning Attacks. In *Advances in Neural Information Processing Systems*, 2017. 16

Guanhong Tao, Guangyu Shen, Yingqi Liu, Shengwei An, Qiuling Xu, Shiqing Ma, Pan Li, and Xiangyu Zhang. Better Trigger Inversion Optimization in Backdoor Scanning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 16

Brandon Tran, Jerry Li, and Aleksander Madry. Spectral Signatures in Backdoor Attacks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 3, 6, 7

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 (IEEE S&P)*, 2019. 16

Ren Wang, Gaoyuan Zhang, Sijia Liu, Pin-Yu Chen, Jinjun Xiong, and Meng Wang. Practical Detection of Trojan Neural Networks: Data-Limited and Data-Free Cases. In *European Conference on Computer Vision*, 2020. 16

Yi Zeng, Won Park, Z. Morley Mao, and Ruoxi Jia. Rethinking the Backdoor Attacks' Triggers: A Frequency Perspective. In *International Conference on Computer Visions (ICCV)*, 2021. 2, 3, 5, 6, 7, 16, 17

Yi Zeng, Si Chen, Won Park, Z. Morley Mao, Ming Jin, and Ruoxi Jia. Adversarial Unlearning of Backdoors via Implicit Hypergradient. In *International Conference on Learning Representations*, 2022. 16

A Connections to Existing Work

Data Attribution: A Brief Overview

The problem of training data attribution (TDA) has been explored using various approaches such as influence functions [Koh and Liang, 2017a, Koh et al., 2019], Shapley value-based estimators [Ghorbani and Zou, 2019], empirical influence computation [Feldman and Zhang, 2020], and predictive datamodels [Park et al., 2023].

Broadly, TDA methods can be categorized into three groups: retraining-based methods, gradient-based methods, and predictive attribution models (see Hammoudeh and Lowd [2024] for a survey). Retraining-based methods systematically retrain models with and without specific training samples and observe changes in the model's outputs [Feldman and Zhang, 2020, Ghorbani and Zou, 2019, Jia et al., 2019]. While these methods yield relatively accurate influence scores, they are computationally prohibitive for moderately large models, as the number of retrains often grows with the size of the training data. Gradient-based methods, such as influence functions [Cook and Weisberg, 1980], are computationally cheaper but often produce less reliable influence estimates for complex models [Basu et al., 2021].

Influence functions approximate the effect of individual training samples on a model's predictions by measuring how a prediction changes when a sample's weight is slightly perturbed. They were introduced to machine learning by Koh and Liang [2017b] and have since been refined [Grosse et al., 2023, Kim et al., 2024, Pruthi et al., 2020]. In data poisoning contexts, Seetharaman et al. [2022] used influence functions to mitigate degradation caused by previously identified poisoned data [Steinhardt et al., 2017]. Building on this, we explore how advanced influence functions like EK-FAC [Grosse et al., 2023] can identify training examples disproportionately contributing to anomalous predictions in poisoned models.

Another approach, predictive data attribution, focuses on predicting model behavior directly based on training data [Ilyas et al., 2022, Park et al., 2023]. While this approach can provide accurate influence estimates, the cost of training predictive models remains a significant limitation.

Unlearning: A Brief Overview

Machine unlearning, first proposed by Cao and Yang [2015], enables ML models to "forget" specific data points by removing their influence. This concept has gained importance with data protection regulations such as GDPR in the EU, which enforce the "right to be forgotten." Ideally, unlearning produces models equivalent to retraining from scratch after excluding the target data [Bourtoule et al., 2021, Cao and Yang, 2015, Gupta et al., 2021]. However, retraining is computationally expensive, leading to the development of approximate unlearning methods [Ginart et al., 2019, Guo et al., 2020, Neel et al., 2021]. These methods are often inspired by concepts from differential privacy, with the relevant $((\epsilon, \delta)$-provable unlearning definition formalized in Sekhari et al. [2021].

Recently, the scope of machine unlearning has expanded beyond privacy to address post-hoc system degradation, such as harmful knowledge removal [Li et al., 2024a] and adversarial attacks [Goel et al., 2024, Pawelczyk et al., 2024, Schoepf et al., 2024]. In corrective unlearning, Pawelczyk et al. [2024] demonstrated the difficulty of mitigating strong poisons like Witches' Brew, while Goel et al. [2024] highlighted challenges when the complete set of manipulated data is unknown. These complexities underscore the inherent difficulty of the setting we address in this work.

Data Poisoning Attacks Data poisoning attacks are a significant threat to ML systems due to their ease of deployment and difficulty in detection. Even minor modifications to training data can lead to successful attacks on models trained on large datasets [Carlini et al., 2024]. In this paper, we consider three forms of data poisoning attacks: a *backdoor attack* [Gu et al., 2019] that adds a small patch in the corner of attacked images and modifies their labels to a target label, a *smooth trigger attack* [Zeng et al., 2021] that adds a trained pattern which is both hard to identify either in raw image domain or frequency domain, and *Witches' Brew* [Geiping et al., 2021], which adds a trained imperceptible pattern on attacked images without modifying labels. Note that the first two attacks modify the victim images' labels, while Witches' Brew is a *clean-label attack*.

Data Poisoning Defences Defenses against data poisoning often involve trigger-pattern reverse engineering using clean data [Dong et al., 2021, Guo et al., 2019, Tao et al., 2022, Wang et al., 2019, 2020]. These methods require additional steps such as input pre-filtering, neuron pruning, or fine-tuning [Chen et al., 2019, Li et al., 2021b, Zeng et al., 2022, **?**]. Other approaches, like

Anti-Backdoor Learning [Li et al., 2021a] and BaDLoss [Alex et al., 2024], necessitate tracking model updates and clean training samples, adding complexity to the defense process.

In contrast, our method requires access only to the trained model and a single poisoned test example, offering a simpler yet effective defense mechanism.

B Experiment Details

B.1 Predefined Set for Image Augmentations

We employ a predefined set of standard image augmentation techniques: Flip, Rotation, Color Jitter, Elastic Transformation, Blur, Inversion, Color Switch, and Random Affine transform. For each transform, one augmentation is randomly selected from this set and applied to the affected test image.

B.2 Attack Methods

The attack target and victim class are chosen at random for each trial. We shall now discuss the details for each attack method below. The relevant code is additionally publicly available in our repository: `https://github.com/andyisokay/delta-influence`.

BadNet For CIFAR datasets, we add a 3×3 checkboard-patterned black patch (pixel values set to zero) at the bottom-right corner of each 32×32 image. For the Imagenette dataset, we utilize a larger square 22×22 black patch to ensure successful injection of the poison. The number of poisoned images varies by dataset: 500 for CIFAR10, 350 for CIFAR100, and 858 for Imagenette.

Smooth Trigger The smooth trigger is generated for each dataset following the algorithm proposed in [Zeng et al., 2021]. The number of poisoned images similarly varies by dataset: 500 for CIFAR10, 125 for CIFAR100 and 300 for Imagenette. Since the poison is more powerful, we are able to poison the model with less number of poisoned samples.

Witches' Brew The adversarial pattern is generated according to the method described in [Geiping et al., 2021]. The number of poisoned images similarly varies by dataset: 500 for CIFAR10, 125 for CIFAR100 and 947 for ImageNette respectively. To ensure successful poisoning of Imagenette, we set we set eps=3, which is twice the value used for CIFAR10 and CIFAR100 (eps=16).

B.3 Hyperparameters for Detection Methods

The hyperparameters are optimized through a grid search process to find the best possible values, following the process from Goel et al. [2024]. Specifically:

ActClust We set the number of components, $n_{comp} = 3$, for all experiments. ActClust is quite robust a method, and we find that a value of 3 performs consistently best across all experiments.

SpecSig SpecSig involves two hyperparameters: the spectral threshold, used to identify significant singular values, and the contribution threshold, used to identify significant data point contributions. SpecSig is sensitive to both parameters. Typically, we select the best spectral threshold by grid search per dataset from the values 4, 6, 8, 10 and the contribution threshold from 7, 9, 11, 13. Higher values indicate a stricter constraint, resulting in fewer detected examples.

FreqDef For datasets with different image sizes, we train a specialized classifier following the methodology described in [Zeng et al., 2021].

EK-FAC We typically begin with a threshold value of 0 and select the best threshold among values (0, 10, 100, 500). Higher threshold values imply stricter filtering constraints, leading to fewer detected examples.

Ours Similar to EK-FAC, starting with a threshold value of 0 is generally effective and search over (0, 1, 5, 10, 100). Lower threshold values and smaller indicate stricter filtering constraints. We additionally hyperparameter search over the tolerance for augmentation flip values 0, 1, 2, 3, with 1 proving to be effective in most cases.

B.4 Hyperparameters for SSD

Among the five unlearning methods considered, SSD is particularly sensitive to hyperparameters but is computationally efficient. This allows for lots of runs to select the optimal unlearning result. For each experiment, we evaluate all possible combinations of two SSD hyperparameters, the weight selection threshold, which controls how protective the selection should be, and the weight dampening constant which defines the level of parameters protection. Specifically, we choose the weight selection threshold from values 2, 10, 50 and the weight dampening constant from 0.01, 0.1, 1.

C Results for Ablating Image-Only and Label-Only Augmentations

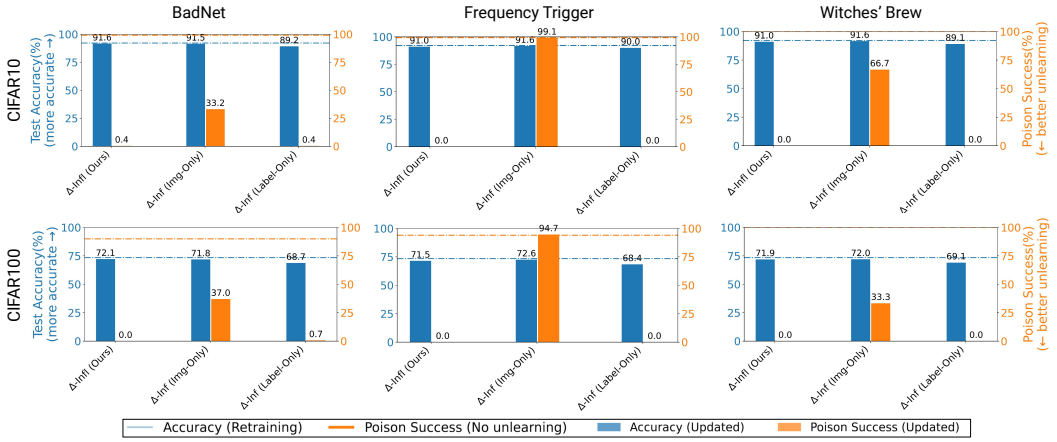

Figure 7: Poison Success Rate and Test Accuracy. This table shows both poison unlearning effectiveness and model utility. A method is considered successful if the poison success rate is below 5%. Label augmentations are instrumental towards identifying poisons, even in the clean-label poison cases. Figure structure from [Pawelczyk et al., 2024].

We show in Figure 7 that Label-Only augmentations are effective in removing the data poisoning (lower poison success rate), while Image-Only augmentations perform poorly in this regard. However, as demonstrated in Table 2, Label-Only augmentations lead to the unnecessary discard of many clean samples, whereas image augmentations significantly reduce the false positive rate, preserving clean data and improving detection precision. Therefore both label and image augmentations are crucial to the effectiveness of the Δ-Influence method.