WHAT DOESN’T KILL YOU MAKES YOU ROBUST(ER): ADVERSARIAL TRAINING AGAINST POISONS AND BACKDOORS

Jonas Geiping, Liam Fowl, Gowthami Somepalli, Micah Goldblum, Michael Moeller, Tom Goldstein
Departments of Computer Science and Mathematics
University of Siegen and University of Maryland
{jonas.geiping,michael.moeller}@uni-siegen.de, gowthami@cs.umd.edu,
{lfowl, goldblum, tomg}@umd.edu

ABSTRACT

Data poisoning is a threat model in which a malicious actor tampers with training data to manipulate outcomes at inference time. A variety of defenses against this threat model have been proposed, but each suffers from at least one of the following flaws: they are easily overcome by adaptive attacks, they severely reduce testing performance, or they cannot generalize to diverse data poisoning threat models. Adversarial training, and its variants, is currently considered the only empirically strong defense against (inference-time) adversarial attacks. In this work, we extend the adversarial training framework to instead defend against (training-time) poisoning and backdoor attacks. Our method desensitizes networks to the effects of poisoning by creating poisons during training and injecting them into training batches. We show that this defense withstands adaptive attacks, generalizes to diverse threat models, and incurs a better performance trade-off than previous defenses.

1 INTRODUCTION

As machine learning systems consume more and more data, the data curation process is increasingly automated and reliant on data from untrusted sources. Breakthroughs in image classification (Rusakovsky et al., 2015) as well as text processing (Brown et al., 2020) are built on large corpora of data scraped from the internet. Automated scraping, in which data is collected directly from online sources, leaves practitioners vulnerable to data poisoning in which bad actors tamper with the data so that models trained on this data perform poorly or contain backdoors embedded in them (Gu et al., 2019; Shafahi et al., 2018). These attacks present security vulnerabilities that persist even if the data is labeled and checked by crowd-sourced human supervision. In essence, entire machine learning pipelines can be compromised if the input data is modified maliciously - even if the modification appears minor and inconspicuous to a human observer. This mounting threat has instilled fear especially in industry practitioners whose business models rely on powerful neural networks trained on massive volumes of scraped data (Kumar et al., 2020).

In response to this growing threat, recent works have proposed a number of defenses against data poisoning attacks (Paudice et al., 2018; Ma et al., 2019). Existing defense strategies suffer from up to three primary shortcomings: 1. In exchange for robustness, they trade off test accuracy to a degree that is intolerable to real-world practitioners (Geiping et al., 2021). 2. They are only robust to specific threat models but not to adaptive attacks specially designed to circumvent the defense (Koh et al., 2018; Tan & Shokri, 2020). 3. They apply only to a specific threat model and do not lend a generally appli-

Figure 1: Data poisoning attacks require a new approach to adversarial training to robustify machine learning models against this threat model.
cable framework to practitioners (Wang et al., 2019). We instead propose a variant of adversarial training that harnesses adversarially poisoned data in the place of (test-time) adversarial examples. We show that this strategy exhibits both an improved robustness-accuracy trade-off as well as greater flexibility for defending against a wide range of threats including adaptive attacks.

Adversarial training desensitizes neural networks to test-time adversarial perturbations by augmenting the training data with on-the-fly crafted adversarial examples (Madry et al., 2018). Similarly, we modify training data in order to desensitize neural networks to the types of perturbations caused by data poisoning - yet adapting this robust training framework to data poisoning requires special consideration of this new threat model. For example, we must decide how to select targets during training in order to simulate targeted data poisoning. We demonstrate the effectiveness of this framework at defending against a range of data tampering threat models including both targeted data poisoning and backdoor trigger attacks on both from-scratch training and transfer learning. We visualize the impact of the defense in feature space and compare to a range of related defense strategies.

2 RELATED WORK

Data poisoning is a class of threat scenarios focused on malicious modifications to the training data of a machine learning model. See Goldblum et al. (2020) for an overview of dataset security. Data poisoning attacks can either focus on denial-of-service attacks on model availability that reduce overall model performance or on backdoor attacks that introduce malicious behavior into an otherwise inconspicuous model which is triggered by a specific visual pattern or target image, thus breaking model integrity (Barreno et al., 2010).

In this work, we focus on attacks against model integrity. In comparison to denial-of-service attacks, which can be noticed before deployment, integrity attacks can insert undetectable backdoors even into models that later pass into production and are used and relied upon in real-world scenarios. These attacks can be further distinguished by the nature of their trigger mechanism. In backdoor trigger attacks (Gu et al., 2019; Turner et al., 2018), the attack is triggered by a specific backdoor pattern or patch that can be added to target images at test time, whereas targeted data poisoning (Shafahi et al., 2018; Zhu et al., 2019) is triggered by a predefined target image. In contrast to targeted poisoning, backdoor trigger attacks can be applied to multiple target images but require target modifications to be active during inference, while targeted attacks are activated by specific, but unmodified targets.

3 GENERALIZING ADVERSARIAL TRAINING TO DATA POISONING

Adversarial training (Madry et al., 2018; Sinha et al., 2018) reduces the impact of test-time adversarial attacks and is generally considered the only strong defense against adversarial examples. Adversarial training solves the saddle-point problem,

$$\min_{\theta} \mathbb{E}_{(x, y) \sim D} \left[ \max_{\Delta \in S} L_{\theta}(x + \Delta, y) \right],$$

where $L_{\theta}$ denotes the loss function of a model with parameters $\theta$, and the adversary perturbs inputs $x$ from a data distribution $D$, subject to the constraint that perturbation $\Delta$ is in $S$. Peri et al. (2020) notes that adversarial training against test-time evasion attacks already confers a small degree of robustness against data poisoning at a performance cost. Our proposed strategy is an adaptation of adversarial training to poisoning, resulting in a stronger defense that degrades performance less than differentially private SGD or adversarial training against evasion attacks. In our adversarial training paradigm, two parties engage in a mini-max game: the attacker maliciously poisons the training data to cause the model to mis-classify targets, while the defender trains the model to correctly classify both poisons and targets. As we describe in the previous section, the capabilities of an attacker depend on its knowledge of the defender’s training setup, so we now enumerate a series of assumptions concerning the knowledge of the attacker and defender before presenting our framework in precise detail.
Table 1: Quantitative result for several attacks and their defense by adversarial poisoning with \( p = 0.75 \) for targeted poisoning and \( p = 0.5 \) for backdoor attacks, showing avg. poison success with standard error (where all trials have equal outcomes, we report the worst-case error estimate 5.59%). The proposed defense significantly decreases success rates over a wide range of attacks and scenarios without any hyperparameter changes.

<table>
<thead>
<tr>
<th>ATTACK</th>
<th>SCENARIO</th>
<th>UNDEFENDED SUCCESS</th>
<th>DEFENDED SUCCESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>BACKDOOR TRIGGERS</td>
<td>FROM-SCRATCH</td>
<td>87.38% (±2.24)</td>
<td>12.93% (±4.59)</td>
</tr>
<tr>
<td>GRADIENT MATCHING</td>
<td>FROM-SCRATCH</td>
<td>90.00% (±6.71)</td>
<td>0.00% (±5.59)</td>
</tr>
<tr>
<td>BULLSEYE POLYTOPE</td>
<td>FINE-TUNING</td>
<td>75.00% (±9.68)</td>
<td>0.00% (±5.59)</td>
</tr>
<tr>
<td>BULLSEYE POLYTOPE</td>
<td>TRANSFER</td>
<td>100.00% (±5.59)</td>
<td>10.00% (±6.71)</td>
</tr>
<tr>
<td>POISON FROGS</td>
<td>TRANSFER</td>
<td>100.00% (±5.59)</td>
<td>15.00% (±7.98)</td>
</tr>
<tr>
<td>GRADIENT MATCHING (SE)</td>
<td>TRANSFER</td>
<td>95.00% (±4.87)</td>
<td>0.00% (±5.59)</td>
</tr>
<tr>
<td>CONVEX POLYTOPE</td>
<td>TRANSFER*</td>
<td>90.00% (±10.00)</td>
<td>40.00% (±16.32)</td>
</tr>
<tr>
<td>HIDDEN TRIGGER BACKDOOR</td>
<td>TRANSFER</td>
<td>55.59% (±5.65)</td>
<td>24.78% (±6.82)</td>
</tr>
</tbody>
</table>

3.1 **Adversarial Poisoning**

Conceptually, poisoning attacks differ from evasion attacks through their intermediacy; the attacker modifies some sample \( x_p \) of the data distribution \( D \) within constraints \( S \), to change model behavior when evaluated on another sample \( x_t \). As such, the defender needs to train to be invariant to any such modifications. Formally, adversarial poisoning thus requires approximating a robust estimation objective given by

\[
\begin{align*}
\min_\theta & \mathbb{E}_{(x_p, y_p) \sim D} \left[ \mathcal{L}_\theta(x_p + \Delta_p, y_p) + \mathcal{L}_\theta(x_t + \Delta_t, y_t) \right] \\
\text{s.t.} & \quad \Delta_p, \Delta_t \in \arg \min_{\Delta_p, \Delta_t \in S} \mathcal{L}_{\text{adv}}(x_p + \Delta_p, x_t + \Delta_t, \theta),
\end{align*}
\]

(2)

where \( \mathcal{L}_\theta \) denotes the loss of a model with parameters \( \theta \), and \( \mathcal{L}_{\text{adv}} \) denotes the objective function of an arbitrary data poisoning attack. \( x_p \) simulates training data (to be poisoned with \( \Delta_p \)) and \( x_t \) simulates a target with possible trigger \( \Delta_t \). For example, in gradient matching, \( \Delta_t \) is zero since such an attack does not modify targets, and \( \Delta_p \in \arg \min_{\Delta_p \in S} \mathbb{V} \mathbb{L}_\theta(x_p + \Delta_p, y_p) \mathbb{V} \mathbb{L}_\theta(x_t, y_t) \) minimizes the cosine similarity, while for simple backdoor triggers \( \Delta_p = \Delta_t \) are non-optimized randomly drawn patches. This robust estimation problem is a strict generalization of adversarial training, which we can recover via \( \mathcal{L}_{\text{adv}} = -\sum_{i \in \{p, t\}} \mathcal{L}_\theta(x_i + \Delta_i, y_i) \).

To realize an approximation to this objective, given each mini-batch of data, we first split this batch into two subsets of data, \( (x_p, y_p) \) and \( (x_t, y_t) \), at random with probability \( p \) for an image to be placed in the poison partition, we then run a chosen data poisoning attack with \( x_p, x_t \), and then train the model on the concatenated output, as seen in fig.1. This way we alternate between both steps in eq. (2) effectively, modifying one part of the batch to indirectly influence the other part and training against this. The full algorithm is summarized in algorithm1.

4 **Experiments**

This section details a quantitative analysis of the proposed defense for the application of image classification with deep neural networks. To fairly evaluate all attacks and defenses, especially in light of Schwarzchild et al. (2020) discussing the difficulty in comparing attacks across different evaluation settings, we implement all attacks and defenses in a single unified framework, which we will make publicly available. For all experiments, we measure avg. poison success over 20 trials, where each trial represents a randomly-chosen attack trigger from a random class and a separately attacked and trained model. The sampling of randomized attack triggers is crucial to estimate the average performance of poisoning attacks, which are generally more effective for related class labels. We discuss additional experimental details in the supp. material.
Figure 2: Avg. Poison Success versus validation accuracy for various defenses against the gradient matching attack of (Geiping et al., 2021) in the from-scratch setting. The baseline undefended model is shown in blue, the proposed defense in red. The differentially private SGD is shown for noise values from 0.0001 to 0.01. The proposed defense provides a strong trade-off of robustness and accuracy.

4.1 Defending in Diverse Scenarios

To evaluate the proposed defense mechanism thoroughly, we first apply the proposed defense against a range of attacks and settings in table 1, choosing \( p = 0.75 \) for all targeted data poisoning attacks and \( p = 0.5 \) for all backdoor attacks and no additional modifications. All attacks shown are adaptive, if possible. In the fine-tuning and transfer scenarios, the pre-trained model is defended but known to the attacker exactly. In all cases, we observe that while the attacks are highly effective against an undefended model, our defense steeply reduces the effectiveness of both poisons and backdoors. These encouraging results suggest that the proposed methodology is a strong strategy that can be robustly applied across a range of attacks and may also be applicable to new settings and attacks proposed in the future.

4.2 Comparison to Other Defenses

In this subsection, we compare the proposed defense to other existing defense strategies against data poisoning including differentially private SGD, adversarial training, various data augmentations, and filter defenses. For differentially private SGD and adversarial training, we test several noise levels and perturbation budgets, respectively. When comparing to filtering defenses, we allow an optimal hyperparameter choice by supplying the exact number of poisons in the training set, although this information would be unknown in practice. We analyze adversarial poisoning with varying levels of \( p \) to show the trade-off of performance and security.

We conduct our comparison in the common from-scratch setting, where the entire model is re-trained. We test the gradient matching attack proposed in Geiping et al. (2021), for a ResNet-18 trained on CIFAR-10 with budget 1% and \( \epsilon = 16 \). While previous defenses were shown to be ineffective in Geiping et al. (2021), we now show in fig. 2 that the proposed adversarial poisoning defense is an extremely effective defense in the from-scratch setting, yielding a much stronger protection than filter defenses, but with only mild trade-off in validation accuracy compared to differential privacy and adversarial training.

5 Conclusions

In this work, we adapt adversarial training to defend against data poisoning and backdoor attacks. In addition to demonstrating the strong defensive capabilities of our method, adversarial poisoning, we analyze the feature space of defended models and observe mechanisms of defense. We stress that we believe this strategy to be a general paradigm for defending against data tampering attacks that can extend to novel future attacks. We are especially interested in understanding this defense better empirically for backdoor attacks as well. Works such as Sun et al. (2020) show that backdoor attacks can fundamentally representations of attacked networks and we wonder if these effects are stabilized by training robustly defended models.
REFERENCES


A APPENDIX

A.1 ADAPTIVE ATTACK SCENARIOS

A crucial step in the design of new defense algorithms is their ability to withstand adaptive attacks, i.e. strong attacks that can be modified to respond to a novel defense algorithm when the attacker is aware of the defense. While this principle has been well-regarded in literature about adversarial attacks at test-time Carlini et al. (2019); Tramer et al. (2020), it has not been applied as rigorously for data poisoning.

The defense proposed in this work is exceedingly effective against non-adaptive models (evaluating the exemplary case of gradient matching), as the difference in training regimes leads to incorrect perturbations computed by the attacker that relies on a pre-trained surrogate model. However, this would also be the case for most modifications to the training procedure, such as adding data augmentations or changing learning rates or optimizer settings. As such, we find that the optimal way to attack this defense is for the attacker to re-train their pre-trained model with exactly the same defense and the same hyperparameters. The attacker can then more accurately estimate the target gradient (for gradient matching) or target features (for feature collision). We also investigated the possibility of applying algorithm 1 during the optimization of poisoned data itself as an additional stochastic input modification. However, this modification weakens the attack by gradient masking, making it too difficult for the attacker to optimize the poisoned data. This behavior mirrors (test-time) adversarial attacks, where it is non-optimal to add additional perturbations during the creation of an adversarial perturbation. As we will find in the analysis section, the defense has a major impact on the feature space of a model, which may make it difficult to bypass the defense with other adaptive attacks.

instances would likely be small. However, poisoned data points in Geiping et al. (2021) are in practice chosen from the same class as the target adversarial label, and this choice can be replicated for the randomly chosen subset of poisoned data points $x_p$ with labels $y_p$ by choosing $y_t$ as the label that appears most often in $y_p$. 


Algorithm 1: Modified iterative training routine.

**Input:** Split probability $p \in (0, 1)$.

repeat
  Sample mini-batch of data $\{x_i, y_i\}_{i=1}^n$.
  Split data randomly into two subsets $x_p, x_t$ with probability $p$.
  Draw malicious labels $y_t$ for $x_t$.
  Apply a data poisoning attack modifying $x_p + \Delta_p$ to reclassify $x_t + \Delta_t$ as $y_t$.
  Concatenate $x_p, x_t$ into a new batch $x_m$.
  Update model based on new data $x_m$.

until training finished

B ANALYSIS

Figure 3: Visualization of the effects of data poisoning attacks via gradient matching against an undefended and a defended model. The target image is marked by a black triangle and is originally part of the class colored blue. The poisoned images are colored red and are part of the class colored green. The $x$-$y$ axis in each diagram corresponds to a projection of the principal direction separating both classes, while the confidence in the original target class is marked on the $z$-axis.

To understand the effect of the proposed adversarial poisoning scheme qualitatively, we conduct an analysis of feature space visualizations.

In fig. 3a, we analyze the defense against the gradient matching attack of Geiping et al. (2021) in the from-scratch setting, where the model is fully re-trained. The attack can be seen to be effective in changing the decision boundary of the model to fit the target without collisions by clustering poisons opposite to the target in feature space, significantly moving the target. However, this is prevented by the defense as seen in fig. 3b. The robust model is not modified by the clustering of poisoned images, and outliers seen in the undefended model are again reclassified as the target class leading to a consistent decision. An interesting side effect of the defense for both attacked and clean models is that the model itself is generally less over-confident in its clean predictions.