DIFFUSION DENOISING AS A CERTIFIED DEFENSE AGAINST CLEAN-LABEL POISONING ATTACKS

Anonymous authors
Paper under double-blind review

ABSTRACT

We present a certified defense to clean-label poisoning attacks. These attacks work by injecting poisoning samples that contain p-norm bounded adversarial perturbations into the training data to induce a targeted misclassification of a test-time input. Inspired by the adversarial robustness achieved by *denoised smoothing*, we show how a pre-trained diffusion model can sanitize the training data before a model training. We extensively test our defense against seven clean-label poisoning attacks and reduce their attack success to 0–16% with only a small drop in the test time accuracy. We compare our defense with existing countermeasures against clean-label poisoning, showing that the defense reduces the attack success the most and offers the best model utility. Our results highlight the need for future work on developing stronger clean-label attacks and using our certified yet practical defense as a strong baseline to evaluate these attacks.

1 INTRODUCTION

A common practice in machine learning is to train models on a large corpus of data. This paradigm empowers many over-parameterized deep-learning models but also makes it challenging to retain the quality of data samples collected from various sources. This makes deep-learning-enabled systems vulnerable to *data poisoning* (Rubinstein et al., 2009; Nelson et al., 2008b; Biggio et al., 2012; Jagielski et al., 2018; Shafahi et al., 2018; Carlini, 2021)—where an adversary can modify a victim model's behaviors by injecting malicious samples (i.e., poisons) into the training data.

But in practice, it may be challenging to make modifications to a training datasets used for fully-supervised classification, because they are often relatively small. As a result, recent work has developed *clean-label poisoning attacks* (Shafahi et al., 2018; Zhu et al., 2019; Aghakhani et al., 2021; Turner et al., 2019; Saha et al., 2020; Huang et al., 2020; Geiping et al., 2021b), where the attacker aims to control the victim model's behavior on a few specific test inputs by injecting poisons that visually appear to be correctly-labeled, but in fact include human-invisible adversarial perturbations.

We introduce a defense against targeted clean-label poisoning attacks inspired by defenses to test-time adversarial examples. Existing poisoning defenses fall into two categories *certified* and *heuristic*. Certified defenses offer provable guarantees, but often significantly decrease the utility of defended models at test-time, making them impractical (Ma et al., 2019; Levine & Feizi, 2020; Wang et al., 2022; Zhang et al., 2022). Heuristic approaches (Suciu et al., 2018; Peri et al., 2020; Hong et al., 2020; Geiping et al., 2021a; Liu et al., 2022) demonstrate their effectiveness against existing attacks in realistic scenarios. However, these defenses rely on unrealistic assumptions, such as the defender knowing the target test-time input (Suciu et al., 2018), or are evaluated against specific poisoning adversaries, leaving them improvable against adaptive attacks or future adversaries.

Contributions. We *first* make two seemingly distant goals closer (i.e., providing a provable defense that *also* works in realistic settings) and present a certified defense against clean-label poisoning. Our defense offers a provable guarantee while minimizing the decrease in clean accuracy. For any p-norm bounded adversarial perturbations to the training data, we ensure a certified accuracy higher than the prior certified defenses. The model trained on the tampered data classifies a subset of test-time input x (or $x + \delta$, where $||\delta||_{\ell_p}$ in clean-label backdoor poisoning) correctly. To achieve this goal, we leverage the recent diffusion probabilistic diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Nichol & Dhariwal, 2021). We use an off-the-shelf diffusion model to *denoise* the entire training data before training a model. In §3, we theoretically show how one achieves a certified accuracy under 2-norm bounded adversarial perturbation to the training data.

Removing adversarial perturbations in the training data before model training, we can decouple the certification process from the training algorithm. *Second*, we leverage this computational property of our defense and present a series of training techniques to alleviate the side-effects of employing certified defenses, i.e., the utility loss of a defended model. To our knowledge, we are the first work that decouples these two processes. We propose the warm-starting (Ash & Adams, 2020)–training a model a few epochs on the tampered data–and to initialize the model parameters using a pre-trained model. In §4, we show they attribute to a better performance compared to existing certified defenses.

Third, we extensively evaluate our defense against seven clean-label poisoning attacks in two different scenarios: training a model from-scratch and transfer-learning. In transfer-learning, our defense reduces the attack success to 0% with a negligible accuracy drop of 0.5% in the certified radius of 0.1 in 2-norm perturbations. We also reduce the attack success to 2–16% when a defender trains a model from scratch. We further compare our defense with six poisoning defenses in the prior work. We show more reduction in the attack success and less accuracy drop than those existing defenses.

Fourth, in §5, we discuss research questions that are important in studying poisoning attacks and defenses but are far-neglected in the prior work. We suggest future work directions to answer them.

2 Preliminaries on clean-label poisoning

Threat model. A clean-label poisoning attacker causes the targeted misclassification of a specific *target* test-time sample $(\mathbf{x}_t, y_t)$ by compromising the training data D_{tr} with clean-label poisoning samples D_p. If a victim trains a model f on the poisoned training data $(D_{tr} \cup D_p)$, the resulting model f^*_θ is likely to misclassify the target instance to the adversarial class y_{adv} while preserving the classification behavior of f^*_θ the same on the clean test-set S. The attacker crafts those *poisons* $(\mathbf{x}_p, y_p) \in D_p$ by first taking a few *base* samples in the same domain $(\mathbf{x}_b, y_b)$ and then adding human-imperceptible perturbations δ, carefully crafted by the attacker and also bound to $||\delta||_{\ell_p} \leq \epsilon$, to them while keeping their labels intact $(y_b = y_p)$. A typical choice of the bound is ℓ_2 or ℓ_∞.

Poisoning as a constrained bi-level optimization. The process of crafting optimal poisoning samples (D^*_p) can be formulated as the constrained bi-level optimization problem:

$$D^*_p = \arg\min_{D_p} \mathcal{L}_{adv}(x_t, y_t; f^*_\theta),$$

where $\mathcal{L}_{adv}(x_t, y_{adv}; f^*_\theta)$ is the adversarial loss function quantifying how accurately a model f^*_θ, trained on the compromised training data, misclassifies a target sample x_t into the class an adversary wants y_{adv}. D_p is the set of poisoning samples we craft, and D^*_p is the resulting optimal poisons.

While minimizing the crafting objective $\mathcal{L}_{adv}(x_t, y_t; f^*_\theta)$, the attacker also trains a model f^*_θ on the compromised training data, which is itself another optimization problem, formulated as follows:

$$f^*_\theta = \arg\min_{\theta} \mathcal{L}_{tr}(D_{tr} \cup D_p, S; \theta),$$

where the typical choice of $\mathcal{L}_{tr}$ is the cross-entropy loss, and S is the clean test-set. Combining both the equations becomes a bi-level optimization: find D_p such that $\mathcal{L}_{adv}$ is minimized after training.

To make the attack inconspicuous, the attacker *constraints* this bi-level optimization by limiting the amount of perturbation $\delta = ||x_p - x_b||$ each poisoning sample can contain to $||\delta|| < \epsilon$.

Existing poisoning attacks. Initial work (Shafahi et al., 2018; Zhu et al., 2019; Aghakhani et al., 2021) minimizes $\mathcal{L}_{adv}$ by crafting poisoning samples that are close to the target in the latent representation space $g(\cdot)$. A typical choice of $g(\cdot)$ is the activation outputs from the penultimate layer of a pre-trained model $f(\cdot)$, publicly available to the attacker from the Internet. The attacks have shown effectiveness in *transfer-learning* scenarios, where $f(\cdot)$ will be fine-tuned on the poisoned training data. The attacker chooses base samples from the target class (x_b, y_{adv}) and craft poisons (x_p, y_{adv}). During fine-tuning, $f(\cdot)$ learns to correctly classify poisons in the target's proximity in the latent representation space, which will classify the target into the class (y_{adv}) the attacker wants.

Recent work focuses on making those poisoning attacks effective when f is trained *from scratch* on the compromised training set. To craft poisoning samples effective in "from scratch" scenarios, the attacker requires to approximate the gradients computed on $\mathcal{L}_{adv}$ that are likely to appear in any models. Huang et al. (2020) addresses this challenge by meta-learning; the poison-crafting process simulates all the possible initialization, intermediate models, and adversarial losses computed on those intermediate models. A follow-up work by Geiping et al. (2021b) alleviates the computational overhead of the prior work by proposing an approximation technique, gradient matching, that aligns the gradients from poisoning samples with those computed on a target.

Defenses against clean-label poisoning. Early work on poisoning defenses focuses on filtering out poisons from the training data. Nelson et al. (2008a) and Suciu et al. (2018) compute the training samples negatively impacting the classification results on targets. But in practice, the defender does not know which test-time samples are the attack targets. Follow-up work (Rubinstein et al., 2009; Peri et al., 2020; Tran et al., 2018) leverages unique statistical properties (*e.g.*, spectral signatures) that can distinguish poisons from the clean data. All these approaches depend on the dataset they use to compute the properties. Recent work (Geiping et al., 2021a; Borgnia et al., 2021; Hong et al., 2020; Liu et al., 2022) thus reduces the dependency by proposing data- or model-agnostic defenses, adapting robust training or differentially-private training to poisoning defenses. While shown effective against existing attacks, those defenses are not *provable*.

A separate line of work addresses this issue by proposing *certifiable* defenses (Levine & Feizi, 2020; Wang et al., 2022; Weber et al., 2023), which guarantee the correct classification of test-time samples when an adversary compromises the training data with the number of poisons k and their perturbation radius of r. (Levine & Feizi, 2020; Wang et al., 2022)'s approach is majority voting, where the defender splits the training data into multiple disjoint subsets, train models on them, and run majority voting of test inputs over those models. (Zhang et al., 2022) proposes BagFlip, a model-agnostic defense that leverages the idea of randomized smoothing studied in adversarial robustness. However, the certification offered by these works has only shown effectiveness against toy datasets like MNIST, while the effectiveness is unknown against practical poisoning attacks, such as clean-label poisoning in CIFAR10. It further hinders the deployment of such defenses as they are computationally expensive to derive such certifications.

Our work proposes a provable yet computationally efficient defense against clean-label poisoning that leverages diffusion-denoising probabilistic models. We will show the certification our defense can offer (§3) and the defense outperforms empirical defenses against existing attacks (§4).

3 A PRACTICAL, CERTIFIED DEFENSE USING DIFFUSION DENOISING

Problem definition. We are interested in certifying the prediction of a model f_θ^* trained on the tampered training data $(D_{tr} \cup D_p)$. To achieve this goal, we first define the perturbation radius r as the ℓ_2 distance between two datasets, computed by taking the sum across ℓ_2 perturbations of all images. The perturbation space P_r^π is the set of all the poisoning datasets D_p obtained by tampering samples with the crafting algorithm π with the perturbation radius of r. Operating within this perturbation space P, for any given test-time sample $(\mathbf{x}_t, y_t)$, can produce a certificate that

$$Pr_{D_p \in P_r^\pi(D_{tr})}\left[f^*_{\theta \leftarrow D_p}(\mathbf{x}_t) = f^*_{\theta \leftarrow D_{tr}}(\mathbf{x}_t)\right] > 1 - \alpha$$

with arbitrarily high probability $1-\alpha$, we guarantee that training on any poisoned dataset $D_p \in P_r^\pi(D_{tr})$ in the perturbation radius r will classify the example $\mathbf{x}_t$ the same way as it was classified on the non-poisoned dataset.

3.1 DENOISING DIFFUSION PROBABILISTIC MODEL

In our work, we use off-the-shelf denoising diffusion probabilistic model (DDPMs) (Sohl-Dickstein et al., 2015; Ho et al., 2020; Nichol & Dhariwal, 2021) to remove adversarial perturbations added to the training data by the clean-label poisoning attacker.

DDPMs are a recent generative model that works by learning the diffusion process of the form $x_t \sim \sqrt{1-\beta_t} \cdot x_{t-1} + \beta_t \cdot \omega_t$, where ω_t is drawn from a standard normal distribution $\mathcal{N}(0, \mathbf{I})$ with x_0 sourced from the actual data distribution, and β_t being fixed (or learned) variance parameters. The above process transforms images from the target data distribution to complete random noise over time, and the reverse process creates images from the data distribution starting with random

Gaussian noise. In image classification, a diffusion model with a fixed time-step $t \in \mathbb{N}^+$ and a fixed schedule samples a noisy version of a training image $x_t \in [-1, 1]^{w \cdot h \cdot c}$ of the form:

$$x_t := \sqrt{\alpha_t} \cdot x + \sqrt{1 - \alpha_t} \cdot \mathcal{N}(0, \mathbf{I}),$$

where α_t is a constant derived from t, which decides the level of noise to be added to the image (the noise increases consistently as t grows). During training, the model minimizes the difference between x and the denoised version of x_t, where x_t is obtained by applying the noise at time-step t.

3.2 Prediction and certification

Now we present practical algorithms for running a prediction with the classifiers trained on a poisoned training set $D_{tr} \cup D_p$, and certifying the robustness of the prediction of a test-time sample $\mathbf{x}$. Our work extends the algorithms in prior work on *randomized smoothing* (Cohen et al., 2019) to clean-label poisoning settings. Our algorithms are shown in Algorithm 1.

We first train many classifiers on noised-then-denoised training datasets, and then return the predictions of these classifiers on the target example x_t, with the algorithm `NoiseTrainClassify`. Then, we follow exactly the standard Randomized Smoothing procedure and produce a certificate with `Certify`, which yields a radius on which this example x_t is guaranteed to be robust.

Pseudocode Noise, Denoise, Train, Classify

```
NoiseTrainClassify($D, f, x, \sigma$)
    $t^*, \alpha_{t^*} \leftarrow$ GetTStep($\sigma$)
    # run denoised smoothing on $x$ and $D$
    $D_{t^*}, x_{t^*} \leftarrow$ Noise($D, x, \sigma$)
    $\hat{D_{t^*}}, \hat{x} \leftarrow$ Denoise($D_{t^*}, x_{t^*}; t^*$)
    $f_{\theta_{t^*}} \leftarrow$ Train($\hat{D_{t^*}}, f$)
    return $f_{\theta_{t^*}}(\hat{x})$

GetTStep($\sigma$)
    $t^* \leftarrow$ find $t$ s.t. $\frac{1-\alpha_t}{\alpha_t} = \sigma^2$
    return $t^*, \alpha_{t^*}$
```

Figure 1: **Prediction and certification** can be implemented with a few (~10) lines of code.

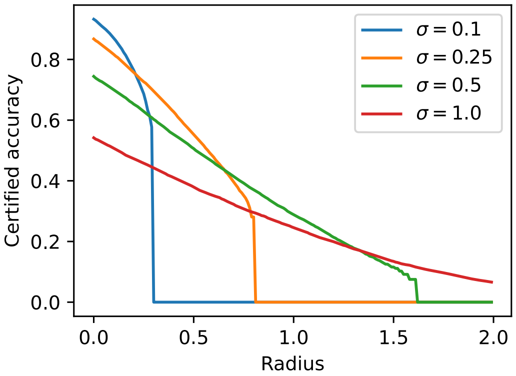

Figure 2: **Certified radius and accuracy** for differing levels of Gaussian noise σ added during training.

Prediction. To compute the prediction for a test-time example, we exactly follow standard randomized smoothing and run `NoiseTrainClassify` algorithm a sufficient number of times (e.g., 1000) and to compute independent predictions for different noised training datasets. The output is then the majority vote label from these classifiers we train.

Certification. Producing a certificate also exactly follows standard randomized smoothing: we first count the number of occurrences of the most likely label compared to any other runner-up label, and from this can derive a radius on which this point is guaranteed to be robust.

Results. We train $10,000$ CIFAR10 models using an efficient training pipeline that trains a single model in 15 seconds on an A100 GPU (for a total of 1.7 GPU days of compute). These models have a clean accuracy of between 92% and 58% depending on the level of noise introduced, and for the first time can certify nontrivial robustness to clean label poisoning attacks on CIFAR-10. A full curve comparing certified accuracy for all given perturbation radii is shown in Figure 2.

3.3 Computational efficiency for certification

In order to obtain a robustness certificate, we must train many CIFAR10 models on different denoisings of the training dataset. This step of the process is computationally expensive, but is in line with the computational complexity of prior randomized smoothing based approaches that require similar work. However, as we will show later, in practice we can empirically obtain robustness with only a very small number of training runs (even as low as **one**!), confirming the related observation from (Lucas et al., 2023) who find a similar effect holds for randomized smoothing of classifiers.

4 Empirical evaluation

Now we empirically evaluate the effectiveness of our certified defense against clean-label poisoning attacks. We utilize the poisoning benchmark developed by Schwarzschild et al. (2021). This benchmarks run clean-label poisoning attacks with the same attack and training configurations, which allows us to compare our defense's effectiveness across multiple attackers.

Poisoning attacks. We evaluate our defense against seven clean-label poisoning attacks: four targeted poisoning (Poison Frogs! (Shafahi et al., 2018), Convex Polytope (Zhu et al., 2019), Bullseye Polytope (Aghakhani et al., 2021), Witches' Brew (Geiping et al., 2021b)) and three backdoor attacks (Label-consistent Backdoor (Turner et al., 2019), Hidden-Trigger Backdoor (Saha et al., 2020), and Sleeper Agent (Souri et al., 2022)). The first three targeted poisoning attacks operate in the transfer-learning scenarios, whereas the rest assumes a victim's model being trained from-scratch. We note that while our defense can offer provable guarantees only to targeted poisoning attacks, we want to test whether the defense can mitigate clean-label backdooring like the prior work did.

Metrics. We consider two metrics: clean accuracy and attack success rate. We compute the accuracy on the entire test set. We consider an attack successful when a poisoned model classifies a target sample to the intended class. When evaluating backdoor attacks, we exclude the cases from the attack success when a tampered model misclassifies the target sample without any trigger pattern.

Methodology. We run our experiments in CIFAR10 (Krizhevsky et al., 2009). Following the standard practices in the prior work (Schwarzschild et al., 2021), we run each attack 100 times on different targets and report the averaged metrics over the 100 runs. We randomly choose 10 targets from each of the 10 CIFAR10 classes. We set the poisoning budget to 1% for all our attacks, e.g., the attacker can tamper with only 500 samples of the entire 50k training instances in CIFAR10.

We evaluate against both the transfer-learning and training from-scratch attacks. In transfer-learning, we fine-tune a pre-trained model on the tampered training set while we train a randomly initialized model in the training from-scratch scenario. We also evaluate our defense against the attackers with two types of knowledge on a victim: white-box and black-box. The white-box attackers know the victim model's architecture (and its parameters), while the black-box attackers do not.

We describe the attack configurations and training hyper-parameters in detail in Appendix A.

4.1 Diffusion denoising renders clean-label poisoning ineffective

We first show to what extent our defense makes the seven poisoning attacks ineffective. Our defense is attack-agnostic. A defender does not know whether the training data contains poisoning samples or not; instead, the defender always trains a model on the denoised training set. We measure the trained model's accuracy and attack success and compare them with the baseline, where a model is trained directly without denoising. We evaluate with different defense strengths: $\sigma \in \{0.1, 0.25, 0.5, 1.0\}$. The results are from attacking with the perturbation bound of 16-pixels ($||\delta||_\infty$=16). Our results against a weaker poisoning attack ($||\delta||_\infty$=8) are shown in Appendix B.

			Our denoising defense at σ (%)				
Poisoning attacks	K	**Scenario**	**[†]0.0**	**0.1**	**0.25**	**0.5**	**1.0**
Witches' Brew (Geiping et al., 2021b)	1	WB	$^{(92.2)}$44.0	$^{(86.5)}$6.0	$^{(72.4)}$4.0	$^{(46.5)}$10.9	$^{(42.1)}$12.1
	4		$^{(92.3)}$65.0	$^{(86.5)}$9.0	$^{(71.9)}$3.0	$^{(46.0)}$9.0	$^{(41.3)}$7.0
Sleeper Agent (Souri et al., 2022)	4		$^{(92.2)}$31.5	$^{(86.0)}$15.7	$^{(72.8)}$10.0	$^{(46.5)}$30.5	$^{(42.9)}$14.1
Witches' Brew (Geiping et al., 2021b)	1	BB	$^{(90.1)}$22.0	$^{(84.6)}$8.1	$^{(70.4)}$4.0	$^{(49.2)}$9.9	$^{(42.6)}$10.3
	4		$^{(90.0)}$33.5	$^{(84.6)}$2.0	$^{(70.1)}$2.0	$^{(49.2)}$8.0	$^{(42.7)}$9.0
Sleeper Agent (Souri et al., 2022)	4		$^{(91.0)}$16.9	$^{(84.5)}$13.1	$^{(69.6)}$10.6	$^{(48.4)}$28.2	$^{(42.7)}$11.1

Table 1: **Defense effectiveness in training from-scratch scenarios.** We measure the accuracy and attack success of the models trained on the denoised training set. Each cell shows the accuracy in the parentheses and the attack success outside. [†] indicates the baseline (no-defense), K is the number of ensembles, and WB and BB stand for the white-box and the black-box attacks, respectively.

Training from-scratch scenarios. Table 1 summarizes our results. Each cell contains the accuracy and the attack success rate of a model trained on the denoised training set. All the numbers are the averaged values over 100 attacks. We demonstrate that our defense mitigates both attacks. In Witches' Brew, the attack success decreases from 44–65% to 6–9%; against Sleeper Agent, the attack success decreases from 32% to 13% at most[1]. We also show that to reduce their attack success to ∼10%, our defense needs at a minimum σ of 0.1–0.25. Increasing σ may defeat a stronger attack but significantly decrease a model's accuracy. This result is interesting, as in our theoretical analysis shown in Sec 3, our defense requires σ of 0.25, while 0.1 is sufficient to defeat both the attacks.

		Our denoising defense at σ (%)				
Poisoning attacks	**Scenario**	†**0.0**	**0.1**	**0.25**	**0.5**	**1.0**
Poison Frog! (Shafahi et al., 2018)	White-box	$^{(93.6)}$69.5	$^{(93.3)}$0.0	$^{(92.6)}$0.0	$^{(90.8)}$0.0	$^{(87.5)}$0.0
Convex Polytope (Zhu et al., 2019)		$^{(93.7)}$12.6	$^{(93.3)}$0.0	$^{(92.6)}$0.0	$^{(90.8)}$1.0	$^{(87.6)}$0.0
Bullseye Polytope (Aghakhani et al., 2021)		$^{(93.5)}$100	$^{(93.3)}$0.0	$^{(92.6)}$0.0	$^{(90.8)}$0.0	$^{(87.5)}$0.0
Label-consistent Backdoor (Turner et al., 2019)		$^{(93.4)}$6.3	$^{(93.3)}$0.0	$^{(92.6)}$0.0	$^{(90.7)}$0.0	$^{(87.4)}$0.0
Hidden Trigger Backdoor (Saha et al., 2020)		$^{(93.2)}$1.0	$^{(93.3)}$0.0	$^{(92.6)}$0.0	$^{(90.8)}$0.0	$^{(87.4)}$0.0
Poison Frog! (Shafahi et al., 2018)	Black-box	$^{(92.2)}$4.2	$^{(91.8)}$0.0	$^{(90.8)}$0.0	$^{(89.3)}$0.0	$^{(86.7)}$0.0
Convex Polytope (Zhu et al., 2019)		$^{(92.3)}$3.1	$^{(91.8)}$0.0	$^{(90.8)}$0.0	$^{(89.3)}$0.0	$^{(86.7)}$0.0
Bullseye Polytope (Aghakhani et al., 2021)		$^{(92.0)}$11.7	$^{(91.8)}$0.0	$^{(90.8)}$0.0	$^{(89.3)}$0.0	$^{(86.7)}$0.0
Label-consistent Backdoor (Turner et al., 2019)		$^{(92.1)}$6.3	$^{(91.8)}$0.0	$^{(90.8)}$0.0	$^{(89.3)}$0.0	$^{(86.7)}$1.9
Hidden Trigger Backdoor (Saha et al., 2020)		$^{(91.5)}$1.0	$^{(91.8)}$0.0	$^{(90.8)}$0.0	$^{(89.3)}$0.0	$^{(86.7)}$1.4

Table 2: **Defense effectiveness in transfer-learning scenarios.** We measure the clean accuracy and attack success of the models trained on the denoised training set. Each cell shows the accuracy in the parentheses and the attack success outside. Note that † indicates the baseline without any denoising.

Transfer-learning scenarios. Table 2 shows our results with the same format as in Table 1. Our defense is also effective in mitigating the attacks in transfer-learning. Against the white-box attacks, we reduce the success rate of the targeted poisoning attacks by 28–90% at σ of 0.1. In the black-box attacks, the targeted poisoning is less effective (with 3–12% attack success), but still, the defense reduces the success to 0%. We observe that the two clean-label backdoor attacks show 1–6% success regardless of the attacker's knowledge. We therefore exclude these two attacks from the subsequent evaluation. Using σ greater than 0.5 increases the attack success slightly. However, this is not because of the successful poisoning attacks but due to the significant decrease in a model's accuracy (5-7%). Here we had the same results: defeating these poisoning attacks requires much smaller σ values (i.e., $\sigma = 0.1$) than what we've seen in our theoretical analysis ($\sigma = 0.25$).

4.2 Improving model utility under our denoising defense

A shortcoming of existing certified defenses (Levine & Feizi, 2020; Wang et al., 2022; Zhang et al., 2022) is that the certified accuracy is substantially lower than the undefended model's accuracy. We also observe in the previous subsection that our defense, when we train a model from scratch on the denoised data, is not free from the same issue, which hinders their deployment in practice.

The theoretical analysis in §3 shows that our defense is agnostic to how we *initialize* a model. We leverage this property and minimize the utility loss by initializing model parameters in specific ways before we train the model on the denoised data. We test two scenarios: a defender can use in-domain data or out-of-domain data to pre-train a model and use its parameters to initialize.

Initializing model parameters using in-domain data. Our strategy here is an adaptation of warm-starting (Ash & Adams, 2020). We first train a model from scratch on the tampered training data for a few epochs to achieve high accuracy. It only requires 5–10 epochs in CIFAR10. We then apply our defense and continue training the model on the denoised training set. We evaluate this strategy against the most effective attacks of each type: Bullseye Polytope (targeted attacks in transfer-learning), Witches' Brew (targeted attacks in training from-scratch), and Sleeper Agent (backdoors).

Table 3 summarizes our results. The middle three rows are the results of leveraging the in-domain data. We train the ResNet18 models on the tampered training set for 10 epochs and continue training

[1] Note that the typical attack success rate of backdoor attacks are ∼90%; Sleeper Agent is a very *weak* attack.

				Our denoising defense at ϵ (%)				
Poisoning attacks	**Pre-training data**	**Model**	**Warm-start**	**†0.0**	**0.1**	**0.25**	**0.5**	**1.0**
Bullseye Polytope	N/A (Baseline)	ResNet18	✗	$^{(93.5)}$100.0	$^{(93.3)}$0.0	$^{(92.6)}$0.0	$^{(90.8)}$0.0	$^{(87.5)}$0.0
Witches' Brew				$^{(92.3)}$65.0	$^{(86.5)}$9.0	$^{(71.9)}$3.0	$^{(46.0)}$9.0	$^{(41.3)}$7.0
Sleeper Agent				$^{(92.2)}$31.5	$^{(86.0)}$15.7	$^{(72.8)}$10.0	$^{(46.5)}$30.5	$^{(42.9)}$14.1
Bullseye Polytope	In-domain Data (CIFAR10, Tampered)	ResNet18	✓	$^{(93.5)}$100.0	$^{(93.3)}$0.0	$^{(92.6)}$1.0	$^{(\mathbf{90.9})}$1.0	$^{(\mathbf{87.7})}$2.0
Witches' Brew				$^{(92.3)}$65.0	$^{(86.5)}$9.0	$^{(\mathbf{72.7})}$5.0	$^{(\mathbf{47.2})}$7.0	$^{(38.0)}$10.0
Sleeper Agent				$^{(92.2)}$31.5	$^{(86.0)}$18.6	$^{(\mathbf{73.3})}$8.8	$^{(\mathbf{48.1})}$9.6	$^{(36.0)}$11.5
Bullseye Polytope	Out-of-domain Data (ImageNet-21k, Clean)	†ResNet18	✓	$^{(84.4)}$8.0	$^{(83.4)}$2.0	$^{(78.4)}$0.0	$^{(67.5)}$6.0	$^{(54.5)}$6.0
Witches' Brew				$^{(92.3)}$65.0	$^{(83.4)}$2.0	$^{(\mathbf{78.4})}$0.0	$^{(\mathbf{67.5})}$6.0	$^{(\mathbf{54.5})}$6.0
Sleeper Agent				$^{(92.4)}$33.2	$^{(\mathbf{86.4})}$14.0	$^{(72.7)}$5.4	$^{(46.8)}$10.1	$^{(38.6)}$11.2

†ResNet18 in Torchvision library; only the latent space dimension differs from the ResNet18 in prior work.

Table 3: **Improving the performance of defended models.** We show the accuracy of defended models and the attack success rate when we employ two techniques for initializing their parameters. The top three rows are the baseline results from Table 1 and 2, and the next two sets of three rows show our results. We highlight the cells showing the accuracy improvements in **bold**.

on the denoised version of the training data for the rest 30 epochs. We show that our strategy (warm-starting) can increase the accuracy of the defended models while defeating clean-label poisoning attacks. Under strong defense guarantees $\sigma > 0.1$, the models have 0.5–2.2% increased accuracy, compared to the baseline, while keeping the attack success $\sim$10%. In $\sigma = 0.1$, we achieve the same accuracy and defense successes. Our strategy can be potentially useful when a defender needs a stronger guarantee, such as against stronger clean-label poisoning future work will develop.

Using models pre-trained on out-of-domain data. Now, instead of running warm-starting on our end, we can also leverage "warm" models available from the legitimate repositories. To evaluate, we take the ResNet18 model pre-trained on ImageNet-21k[2]. We use this pre-trained model in two practical ways. We take the model as-is and train it on the denoised training data. In the second approach, we combine the first approach with the previous idea. We train the model on the tampered training data for a few epochs and then train this fine-tuned model on the denoised data.

The bottom three rows of Table 3 are the results of using the warm model. We fine-tune the ResNet18 for 40 epochs on the denoised training data. Surprisingly, our second strategy can further improve the accuracy of the final models trained with strong defense guarantees (e.g., $\sigma > 0.1$). The final models achieve 6.5–13.2% greater accuracy than the results shown in Table 1 and 2. In $\sigma = 1.0$, the final accuracy of defended models has a negligible difference from the baseline. We are the first work to offer practical strategies to manage the utility-security trade-off in certified defense.

4.3 Comparison to Existing Poisoning Defenses

We finally examine how our defense works better/worse than existing poisoning defenses. We compare ours with six state-of-the-art defenses: DP (Ma et al., 2019; Hong et al., 2020), DPA (Levine & Feizi, 2020; Wang et al., 2022), BagFlip (Zhang et al., 2022), ROE (Rezaei et al., 2023), k-NN (Peri et al., 2020), AP (Geiping et al., 2021a), and FrieNDs (Liu et al., 2022). The first four are certified defenses like ours, and the last three are non-certified ones. Here we test them in the same attack settings we use in the previous sections (e.g., $||\delta||_\infty = 16$ and in CIFAR10).

We first start with comparing our defense with approaches to filtering out poisoning samples.

Poisoning attacks	**TP**	**FP**	**k-NN**	**Ours**
Witches' Brew	31	464	$^{(86.3)}$3.0	$^{(86.5)}$9.0
Bullseye Polytope	431	63	$^{(93.7)}$24.2	$^{(93.3)}$0.0

Table 4: **Comparison to Deep k-NN.** Our defense is better than Deep k-NN (Peri et al., 2020) at mitigating clean-label poisoning attacks.

Comparison to Deep k-NN. Peri et al. (2020) remove poisons from the training data by leveraging the k-nearest-neighbor (k-NN) algorithm. They run k-NN on the latent representations of the training data to identify potentially malicious samples (i.e., samples with labels different from the nearest ones) and remove them. We compare Deep k-NN's effectiveness to our defense. We use the defense configurations that bring the best results in the original study, i.e.,

[2] https://pytorch.org/vision/stable/models/generated/torchvision.models.resnet18.html

setting k to 500. Table 5 summarizes our results. Deep k-NN fails to remove most poisoning samples (but it removes many benign samples!). Our defense achieves a higher success in defeating clean-label poisoning attacks.

Now, we turn our attention to the training-time defenses and compare ours with them.

	Heuristic defense			Certified defense	
Poisoning attacks	**DP-SGD**	**AT**	**FrieNDs**	**ROE**	**Ours**
Bullseye Polytope (Aghakhani et al., 2021)	$^{(93.9)}7.0$	$^{(90.3)}96.4$	$^{(90.2)}10.0$	*N/A	$^{(93.5)}0.0$
Witches' Brew (Geiping et al., 2021b)	$^{(76.4)}4.0$	$^{(66.3)}2.4$	$^{(87.6)}8.0$	$^{(70.2)}10.0$	$^{(86.5)}9.0$
Sleeper Agent (Souri et al., 2022)	$^{(74.7)}2.3$	$^{(66.3)}7.2$	$^{(87.4)}13.6$	$^{(68.6)}12.0$	$^{(86.0)}15.7$

*ROE trains 250 models from scratch on each partition of the training data; incompatible with BP.

Table 5: **Comparison to training-time defenses.** We compare with three heuristic defenses and one recent certified defense against clean-label poisoning attacks most effective in our experiments.

vs. DP-SGD. Ma et al. (2019) proposed a certified defense against poisoning attacks that leverage differential privacy (DP). They make a model less sensitive to a single-sample modification to the training data. But the defense offers a weak guarantee in practice; for example, the certificate is only valid in CIFAR10, when an adversary tampers one training sample. Hong et al. (2020) later empirically shows that DP is still somewhat effective against poisoning. We compare our defense to the training with $(1.0, 0.05)$-DP[3] on the same tampered training data. We use Opacus[4] to train models with DP and keep all the other configurations the same. In Table 5, both defenses significantly reduce the attack success to 0–16%. Our defense achieves at most 10% higher accuracy than DP. DP reduces the attack success slightly more against clean-label backdooring.

vs. AT. Geiping et al. (2021a) adapts the adversarial training (AT) (Madry et al., 2017) for defeating clean-label poisoning: in each mini-batch, instead of crafting adversarial examples, they synthesize poisoning samples and have a model to make correct classifications on them. While shown some effectiveness, this approach could overfit a specific poisoning attack considered in the adapted AT, leaving the model vulnerable to unseen poisoning attacks. We therefore compare our defense with the original AT with the PGD-7 bounded to ϵ of 4. Table 5 shows that our defense achieves higher accuracy than the robust models while making the attacks ineffective. Interestingly, AT cannot defeat the BP attacks. We hypothesize that clean-label poisons in the training data are already adversarial examples; thus, PGD-7 may not add more perturbations to these poisons during training.

vs. FrieNDs. Liu et al. (2022) advances the idea of training robust models. Instead of adding random Gaussian noise to the data during training, they use "friendly" noise, pre-computed by a defender, that minimizes the accuracy loss to a model they will train. When their defense is in action, they train a model on the tampered training data with two types of noise: the friendly noise computed before and a weak random noise sampled uniformly within a bound. We compare our defense with FrieNDs, and both greatly reduce the attack's success while preserving the model's utility. The critical difference is that ours is a certified defense, while FrieNDs is not. A promising future work we leave here is developing adaptive clean-label poisoning attacks against FrieNDs.

vs. ROE. Certified defenses against data poisoning attacks (Steinhardt et al., 2017; Ma et al., 2019; Diakonikolas et al., 2019; Levine & Feizi, 2020; Gao et al., 2021; Zhang et al., 2022; Wang et al., 2022; Rezaei et al., 2023) are; however, we found that most of these defenses showed their effectiveness in limited scenarios, such as in MINST-17 (Zhang et al., 2022), a subset of MNIST containing only the samples of the digits 1 and 7. In consequence, they aren't compatible with our clean-label poisoning settings. A recent defense by Rezaei et al. (2023) demonstrates the certified robustness in practical settings; we thus compare our defense to their run-off-election (ROE). In Table 5, we empirically show that our defense is more effective than ROE while minimizing the accuracy drop.

Comparison of computational overhead. Most training-time defenses require additional mechanisms we need to "add" to the training algorithm. AT needs to craft adversarial examples (or clean-label poisons (Geiping et al., 2021a)). DP-SGD introduces noise into the gradients during training. FrieNDs pre-computes friendly noise and then optimally applies the noise during the training, and

[3] We set the clipping norm and noise multiplier to 1.0 and 0.05, following the prior work (Lecuyer et al., 2019).

[4] https://opacus.ai

ROE requires training of 250 base classifiers to achieve the improved certificates. In contrast, our defense only requires to run forward *once* with an off-the-shelf diffusion model.

5 Discussion and Future Work

Our work offers a new perspective on the cat-and-mouse game played in clean-label poisoning.

Initial claims from the cat: Clean label poisoning attacks, with human-imperceptible perturbations that keeping the original labels, are an attractive option to inject malicious behaviors into models. Even if a victim carefully curates the training data, these attacks remain effective by evading *filtering* defenses that rely on statistical signatures distinguishing clean samples from poisons (Nelson et al., 2008a; Suciu et al., 2018; Peri et al., 2020).

The mouse's counterarguments: Yet, as seen in research on the adversarial robustness, such small perturbations can be *brittle* (Carlini et al., 2022): by adding a large quantity of noise and then denoising the images, the adversarial perturbations become nearly removed—and so clean-label perturbations are also removed. In prior work (and in this work as well), we leverage this observation and propose both certified and non-certified heuristic defenses.

Additional arguments from the cat: A counterargument from the clean-label attack's side was that those defenses inevitably compromise a model's performance—a fact corroborated by the prior work on the adversarial robustness (Tsipras et al., 2018; Zhang et al., 2019). Similarly, defenses could greatly reduce the poisoning attack success, but at the same time, they decrease the classification accuracy of defended models, often tipping the scales in favor of the adversary. If a defense yields a CIFAR-10 model with 60–70% utility, what is the point of having such models in practice? Indeed, our own certified models require degraded utility to achieve certified accuracy. And other defenses, e.g., differentially-private training (Ma et al., 2019; Hong et al., 2020), are computationally demanding, increasing the training time of a model by an order of magnitude.

Our work shows that this cat-and-mouse game need not be as pessimistic as the cat portrays.

By leveraging an off-the-shelf diffusion denoising model (Ho et al., 2020; Nichol & Dhariwal, 2021), we can purify the training data, possibly containing malicious samples, *offline* and render six existing clean-label attacks ineffective. With a small provable guarantee against the training data contamination ($||\epsilon||_\infty = 2$), the majority of attacks reach to 0% success. A few recent attacks exhibit 0–10% success rate in CIFAR10—a rate comparable to random misclassification of test-time inputs.

In contrast to the cat's counterargument, we can also address the accuracy degradation problem—as long as we do not need certified robustness. Existing defenses that work to defend against poisoning require applying a particular training algorithm designed by the defender (Hong et al., 2020; Geiping et al., 2021a; Liu et al., 2022; Levine & Feizi, 2020; Wang et al., 2022; Zhang et al., 2022). By decoupling the training process and the defense, we open a new opportunity for a defender to develop training techniques that can enhance the accuracy of the defended models while maintaining robustness. Our experimental results propose two effective heuristic algorithms in improving defended models' utility, especially when the models are under strong defense guarantees (i.e., $||\epsilon||_\infty > 2$).

Next steps? One of our main conclusions—other than that our defense is effective—is that there is an extremely large gap between the certified accuracy and best practical attack accuracy. When using diffusion models to denoise adversarial examples, a practical adversary can achieve within a few percentage points of the worst-case certified result; for example, if we can certify a lower bound of 50% robust accuracy, a practical adversary could achieve 55% robust accuracy. But in this setting, even at perturbation levels where we can't certify *anything*, the attacks fail to decrease the model utility at all. This implies either that (1) current attacks are *not fully exploiting* the space of possible attacks, or (2) that the certificate provided is loose and a more careful analysis could tighten it.

In addition, when studying stronger clean-label poisoning attacks, we encourage future work to include our defense as a strong baseline. We show in §4 that the defense renders the state-of-the-art clean-label attacks ineffective and is, by far, the most effective defense among the existing ones. We believe that strong attacks future work will develop should at least survive against this defense. Our defense is also designed to run without any additional training costs—with an off-the-shelf diffusion model, one can easily denoise the tampered training data and run the training of a model on it.

REFERENCES

Hojjat Aghakhani, Dongyu Meng, Yu-Xiang Wang, Christopher Kruegel, and Giovanni Vigna. Bullseye polytope: A scalable clean-label poisoning attack with improved transferability. In *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 159–178. IEEE, 2021.

Jordan Ash and Ryan P Adams. On warm-starting neural network training. *Advances in neural information processing systems*, 33:3884–3894, 2020.

Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. *arXiv preprint arXiv:1206.6389*, 2012.

Eitan Borgnia, Jonas Geiping, Valeriia Cherepanova, Liam Fowl, Arjun Gupta, Amin Ghiasi, Furong Huang, Micah Goldblum, and Tom Goldstein. Dp-instahide: Provably defusing poisoning and backdoor attacks with differentially private data augmentations. *arXiv preprint arXiv:2103.02079*, 2021.

Nicholas Carlini. Poisoning the unlabeled dataset of {Semi-Supervised} learning. In *30th USENIX Security Symposium (USENIX Security 21)*, pp. 1577–1592, 2021.

Nicholas Carlini, Florian Tramer, Krishnamurthy Dj Dvijotham, Leslie Rice, Mingjie Sun, and J Zico Kolter. (certified!!) adversarial robustness for free! *arXiv preprint arXiv:2206.10550*, 2022.

Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *international conference on machine learning*, pp. 1310–1320. PMLR, 2019.

Ilias Diakonikolas, Gautam Kamath, Daniel Kane, Jerry Li, Jacob Steinhardt, and Alistair Stewart. Sever: A robust meta-algorithm for stochastic optimization. In *International Conference on Machine Learning*, pp. 1596–1606. PMLR, 2019.

Ji Gao, Amin Karbasi, and Mohammad Mahmoody. Learning and certification under instance-targeted poisoning. In *Uncertainty in Artificial Intelligence*, pp. 2135–2145. PMLR, 2021.

Jonas Geiping, Liam Fowl, Gowthami Somepalli, Micah Goldblum, Michael Moeller, and Tom Goldstein. What doesn't kill you makes you robust (er): How to adversarially train against data poisoning. *arXiv preprint arXiv:2102.13624*, 2021a.

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*, 2021b. URL https://openreview.net/forum?id=01olnfLIbD.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Sanghyun Hong, Varun Chandrasekaran, Yiğitcan Kaya, Tudor Dumitraş, and Nicolas Papernot. On the effectiveness of mitigating data poisoning attacks with gradient shaping. *arXiv preprint arXiv:2002.11497*, 2020.

W Ronny Huang, Jonas Geiping, Liam Fowl, Gavin Taylor, and Tom Goldstein. Metapoison: Practical general-purpose clean-label data poisoning. *Advances in Neural Information Processing Systems*, 33:12080–12091, 2020.

Matthew Jagielski, Alina Oprea, Battista Biggio, Chang Liu, Cristina Nita-Rotaru, and Bo Li. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In *2018 IEEE symposium on security and privacy (SP)*, pp. 19–35. IEEE, 2018.

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

Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE symposium on security and privacy (SP)*, pp. 656–672. IEEE, 2019.

Alexander Levine and Soheil Feizi. Deep partition aggregation: Provable defense against general poisoning attacks. *arXiv preprint arXiv:2006.14768*, 2020.

Tian Yu Liu, Yu Yang, and Baharan Mirzasoleiman. Friendly noise against adversarial noise: a powerful defense against data poisoning attack. *Advances in Neural Information Processing Systems*, 35:11947–11959, 2022.

Keane Lucas, Matthew Jagielski, Florian Tramèr, Lujo Bauer, and Nicholas Carlini. Randomness in ml defenses helps persistent attackers and hinders evaluators. *arXiv preprint arXiv:2302.13464*, 2023.

Yuzhe Ma, Xiaojin Zhu, and Justin Hsu. Data poisoning against differentially-private learners: Attacks and defenses. *arXiv preprint arXiv:1903.09860*, 2019.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Blaine Nelson, Marco Barreno, Fuching Jack Chi, Anthony D Joseph, Benjamin IP Rubinstein, Udam Saini, Charles Sutton, J Doug Tygar, and Kai Xia. Exploiting machine learning to subvert your spam filter. *LEET*, 8(1-9):16–17, 2008a.

Blaine Nelson, Marco Barreno, Fuching Jack Chi, Anthony D Joseph, Benjamin IP Rubinstein, Udam Saini, Charles Sutton, J Doug Tygar, and Kai Xia. Exploiting machine learning to subvert your spam filter. *LEET*, 8(1-9):16–17, 2008b.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pp. 8162–8171. PMLR, 2021.

Neehar Peri, Neal Gupta, W Ronny Huang, Liam Fowl, Chen Zhu, Soheil Feizi, Tom Goldstein, and John P Dickerson. Deep k-nn defense against clean-label data poisoning attacks. In *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pp. 55–70. Springer, 2020.

Keivan Rezaei, Kiarash Banihashem, Atoosa Chegini, and Soheil Feizi. Run-off election: Improved provable defense against data poisoning attacks. *arXiv preprint arXiv:2302.02300*, 2023.

Benjamin IP Rubinstein, Blaine Nelson, Ling Huang, Anthony D Joseph, Shing-hon Lau, Satish Rao, Nina Taft, and J Doug Tygar. Antidote: understanding and defending against poisoning of anomaly detectors. In *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement*, pp. 1–14, 2009.

Aniruddha Saha, Akshayvarun Subramanya, and Hamed Pirsiavash. Hidden trigger backdoor attacks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 11957–11965, 2020.

Avi Schwarzschild, Micah Goldblum, Arjun Gupta, John P Dickerson, and Tom Goldstein. Just how toxic is data poisoning? a unified benchmark for backdoor and data poisoning attacks. In *International Conference on Machine Learning*, pp. 9389–9398. PMLR, 2021.

Ali Shafahi, W Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. Poison frogs! targeted clean-label poisoning attacks on neural networks. *Advances in neural information processing systems*, 31, 2018.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. PMLR, 2015.

Hossein Souri, Liam Fowl, Rama Chellappa, Micah Goldblum, and Tom Goldstein. Sleeper agent: Scalable hidden trigger backdoors for neural networks trained from scratch. *Advances in Neural Information Processing Systems*, 35:19165–19178, 2022.

Jacob Steinhardt, Pang Wei W Koh, and Percy S Liang. Certified defenses for data poisoning attacks. *Advances in neural information processing systems*, 30, 2017.

Octavian Suciu, Radu Marginean, Yigitcan Kaya, Hal Daume III, and Tudor Dumitras. When does machine learning FAIL? generalized transferability for evasion and poisoning attacks. In *27th USENIX Security Symposium (USENIX Security 18)*, pp. 1299–1316, Baltimore, MD, August 2018. USENIX Association. ISBN 978-1-939133-04-5. URL https://www.usenix.org/conference/usenixsecurity18/presentation/suciu.

Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. *Advances in neural information processing systems*, 31, 2018.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018.

Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Label-consistent backdoor attacks. *arXiv preprint arXiv:1912.02771*, 2019.

Wenxiao Wang, Alexander J Levine, and Soheil Feizi. Improved certified defenses against data poisoning with (deterministic) finite aggregation. In *International Conference on Machine Learning*, pp. 22769–22783. PMLR, 2022.

Maurice Weber, Xiaojun Xu, Bojan Karlaš, Ce Zhang, and Bo Li. Rab: Provable robustness against backdoor attacks. In *2023 IEEE Symposium on Security and Privacy (SP)*, pp. 1311–1328. IEEE, 2023.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pp. 7472–7482. PMLR, 2019.

Yuhao Zhang, Aws Albarghouthi, and Loris D'Antoni. Bagflip: A certified defense against data poisoning. *Advances in Neural Information Processing Systems*, 35:31474–31483, 2022.

Chen Zhu, W Ronny Huang, Li Hengduo Shafahi, Ali and, Gavin Taylor, Christoph Studer, and Tom Goldstein. Transferable clean-label poisoning attacks on deep neural nets. In *International Conference on Machine Learning*, pp. 7614–7623, 2019.