DOES ADVERSARIAL ROBUSTNESS REALLY IMPLY BACKDOOR VULNERABILITY?

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent research has revealed a trade-off between the robustness against adversarial attacks and backdoor attacks. Specifically, with the increasing adversarial robustness obtained through adversarial training, the model easily memorizes the malicious behaviors embedded in poisoned data and becomes more vulnerable to backdoor attacks. Meanwhile, some studies have demonstrated that adversarial training can somewhat mitigate the effect of poisoned data during training. This paper revisits the trade-off and raises a question *whether adversarial robustness really implies backdoor vulnerability.* Based on thorough experiments, we find that such trade-off ignores the interactions between the perturbation budget of adversarial training and the magnitude of the backdoor trigger. Indeed, an adversarially trained model is capable of achieving backdoor robustness as long as the perturbation budget surpasses the trigger magnitude, while it is vulnerable to backdoor attacks only for adversarial training with a small perturbation budget. To always mitigate the backdoor vulnerability, we propose an adversarial-training based detection strategy and a general pipeline against backdoor attacks, which consistently brings backdoor robustness regardless of the perturbation budget.

1 INTRODUCTION

Recently, deep neural networks (DNNs) have achieved great success in computer vision (He et al., 2016), natural language processing (Devlin et al., 2019), and gaming agents (Silver et al., 2016). Since DNNs are gradually deployed in many safety-critical applications such as autonomous driving (Ding et al., 2019) and smart healthcare (Ali et al., 2020), the security threats to them have aroused tremendous attention (Papernot et al., 2018; Pitropakis et al., 2019; Hu et al., 2021). Among them, adversarial attacks and backdoor attacks are remarkably dangerous. For example, it is well known that DNNs are inherently vulnerable to adversarial examples (Szegedy et al., 2014; Goodfellow et al., 2015), *i.e.*, natural inputs superimposed with intentional and human-imperceptible perturbations to lead to misclassification even with high confidence. Meanwhile, DNNs may learn some backdoor behaviors (Gu et al., 2017; Chen et al., 2017), *i.e.*, always predicting a target label in the presence of a predefined trigger pattern, from poisoned samples hidden in training data (Goldblum et al., 2020) collected from Internet.

Since DNN-based systems may face security threats at any time, *e.g.*, backdoor attacks in training time and adversarial attacks in testing time, it is urgent to obtain robustness against all potential attacks. We illustrate both types of threats with a toy example in Figure 1. Unfortunately, Weng et al. (2020) indicated there exists a trade-off between adversarial robustness and backdoor robustness. Specifically, with the increasing adversarial robustness, adversarially trained DNNs easily memorize the backdoor behaviors embedded in poisoned data and become more vulnerable to backdoor attacks compared to the normally trained ones. However, recent studies indicated that adversarial training (AT) can mitigate the impact from some malicious or corrupted data, which seems to conflict with such a trade-off. For example, Peri et al. (2020) claimed that adversarially trained feature extractors yield robust features resistant against clean-label data poisoning attacks such as feature collision attacks (Shafahi et al., 2018) and convex polytope attacks (Zhu et al., 2019) in the transfer learning. Zhu et al. (2021) indicated that AT encourages DNNs to be locally constant in the neighborhood of correct data and prevents them from memorizing the corrupted labels. Thus, this paper raises the following questions: *Does adversarial robustness really imply backdoor vulnerability? If not, under what condition will the trade-off exist?*

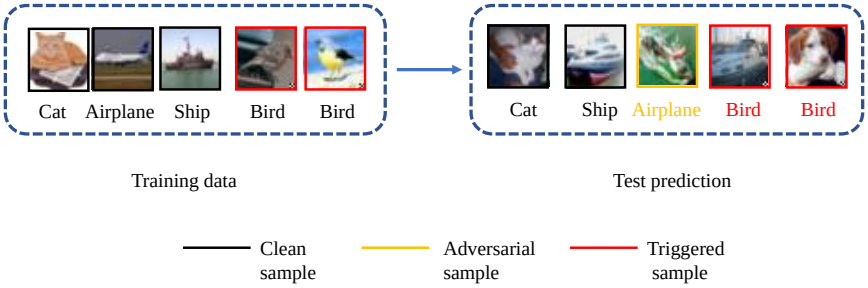


Figure 1: A toy example to illustrate the security threats when encountered with backdoor attacks and adversarial attacks. An adversary performs backdoor attacks by planting a predefined trigger (a checkerboard pattern at the bottom right) in a small portion of training samples in bird class. The test samples will be wrongly classified as birds whenever the presence of the trigger. Meanwhile, the adversary performs adversarial attacks by adding imperceptible perturbations to mislead the model to classify the ship picture as the airplane class.

This paper revisits the trade-off between adversarial robustness and backdoor robustness to answer the questions. We conduct extensive experiments across different settings (including varying poisoning rates and types, trigger shapes and sizes, and architectures) and find that such a trade-off ignores the intersection between the perturbation budget of AT (*e.g.*, the maximum perturbation size ϵ in the l_∞ threat model) and the magnitude of the trigger (*e.g.*, the transparency of a predefined trigger). The trade-off only occurs when the perturbation budget of AT is relatively small and the magnitude of the trigger is large enough. In fact, as long as the perturbation budget of AT surpasses the trigger magnitude, AT can prevent models from learning the backdoor behaviors from poisoned data and improve backdoor robustness. Further, to ensure backdoor robustness regardless of the perturbation budgets in AT, we explore the geometric property of training data when AT results in backdoor vulnerability (smaller perturbation budgets). We find it requires more steps to generate adversarial example from poisoned data using projected gradient descent (PGD) to fool the model, which indicates that poisoned data are farther away from the decision boundary than clean data. Inspired by such findings, we propose a novel backdoor detection method termed *PGD defense*, which effectively distinguishes between poisoned data and clean data. In conclusion, the contributions of this paper can be summarized as follows:

- We empirically demonstrate that the trade-off claim between backdoor and adversarial robustness is not strictly correct and investigate the specific conditions under which adversarial training will benefit or hurt backdoor robustness.
- We propose a backdoor detection method enlightened with the geometric property of training data in adversarial training and demonstrate the proposed method in comparison to other baseline detection methods.
- We propose a general pipeline of adversarial training against backdoor attacks and achieve both high adversarial and backdoor robustness at the same time .

2 Background

2.1 Adversarial Training

We consider a K-class classification problem on the d-dimensional input space and denote $f_\theta : \mathbb{R}^d \to \{0,1\}^K$ as the target function to be optimized. Given a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ and a loss function $\mathcal{L}$ (usually the cross entropy loss), AT optimizes the adversarial loss which is defined as the worst case in the l_p-ball centered at the original training data. The learning objective is

$$\min_\theta \sum_{i=1}^{n} \max_{x_i' \in \mathcal{B}_p(x_i,\epsilon)} \mathcal{L}(f_\theta(x_i'), y_i), \tag{1}$$

in which $\mathcal{B}_p(x_i, \epsilon) := \left\{x_i' \in \mathbb{R}^d : ||x_i' - x_i||_p \leq \epsilon\right\}$. To approximately solve the inner maximization problem, Madry et al. (2018) adopted PGD on the negative loss function:

$$x_i^{(t)} = \Pi_{\mathcal{B}_p(x_i^{(t-1)}, \epsilon)} \left[x_i^{(t-1)} + \beta \cdot \text{sign}(\bigtriangledown_x \mathcal{L}(f_\theta(x_i^{(t-1)}), y_i))\right], \tag{2}$$

where Π denotes the projection onto the convex set $\mathcal{B}_p(x_i^{(t-1)}, \epsilon)$, β is the step size of each iteration step and t is the current iteration step.

A corpus of follow-up work has been proposed to improve the standard AT from their own perspectives and demonstrated the effectiveness (Zhang et al., 2019; Qin et al., 2019; Carmon et al., 2019; Zhang et al., 2020; Wu et al., 2020). Also, another stream of works explored the potential benefits of AT beyond l_p-ball robustness (Peri et al., 2020; Kim et al., 2020; Salman et al., 2020; Xie et al., 2020; Kireev et al., 2021; Zhu et al., 2021). Our work belongs to the second category. Specifically, we study how we can benefit backdoor robustness using AT.

2.2 Backdoor Attack

Accessible to the benign samples used for model training, an adversary performs backdoor attacks by poisoning a small portion of training data so that the presence of a backdoor trigger will elicit the model's specific predictions as the adversary expects (Goldblum et al., 2020). The poisoned sample $\tilde{x}$ is generated according to the following formula:

$$\tilde{x} = (1 - m) \odot [(1 - \alpha)x + \alpha \cdot p] + m \odot x, \tag{3}$$

where $\odot$ is the element-wise multiplication, $p \in \mathbb{R}^d$ is the predefined trigger pattern, $m \in \{0, 1\}^d$ is a binary mask deciding the trigger injecting region and $\alpha \in (0, 1]$ is a transparency parameter concerned with the visibility of the trigger pattern.

Based on the primary backdoor attack, several improvements have been proposed to enhance the effectiveness and stealthiness, such as invisible backdoor attacks (Chen et al., 2017; Turner et al., 2019; Alayrac et al., 2019; Saha et al., 2020; Ning et al., 2021) and sample-specific attacks (Nguyen & Tran, 2020; Li et al., 2020b). To the best of our knowledge, all these works adopt standard training to obtain victim models. Our work intends to investigate the effect of adversarial training on backdoor attacks.

2.3 Backdoor Defense

One family of approaches to alleviate backdoor attacks is identifying backdoor models with trigger reconstruction (Wang et al., 2019a; Chen et al., 2019; Guo et al., 2019; Wang et al., 2020). These methods draw inspirations from the observations that the perturbation cost for perturbing an image to the target class is the smallest out of all labels. Another strategy tackles the threat by removing backdoor behavior from the already trained victim models, such as pruning neurons that are dormant on clean inputs (Liu et al., 2018) or fine-tuning the model on a clean dataset (Chen et al., 2021; Liu et al., 2021).

In this paper, we mainly focus a more direct strategy to eliminate backdoor attacks by identifying and removing the triggered data in training samples. The simplest way to achieve this is outlier detection in the input space (Steinhardt et al., 2017; Diakonikolas et al., 2019). While these methods are effective in low-dimensional domains, they may fail in complex domains, *e.g.*, an image for a pixel space can not convey enough semantic information (Goldblum et al., 2020). Recent works have explored the latent feature representations and improve the detection performance. Chen et al. (2018) analyzed the neural activation and proposed a clustering method to identify the corrupted inputs. Tran et al. (2018) utilized the spectral signatures from learned representations to detect and remove the poisoned data. We will show that our work leads to a novel detection strategy based on AT.

3 EVALUATION OF BACKDOOR VULNERABILITY UNDER AT

In this section, we conduct extensive experiments to explore how adversarial training (AT) impacts on backdoor robustness. Here, we consider the common cases where the adversary only has access to poison some training data[1], and the poisoning paradigm can be found in Eq. (3).

The Limitations of Previous Research. Although Weng et al. (2020) indicated a trade-off between adversarial robustness and backdoor robustness, their findings are only based on experiments which perform adversarial training with a fixed perturbation budget ($\epsilon = 8/255$ on CIFAR-10 under l_∞) and a fixed magnitude of the trigger (the trigger transparency 1.0). Note that the perturbation budget of AT represents the prediction invariance level within the neighbour of clean data, while the magnitude of the trigger represents the perturbation size applied to natural data to lead to misclassification by the adversary. Thus, the intersection between them matters in robustness obtained after training.

(a) Clean (b) 3×3 checkerboard ($\alpha = 0.2$) (c) 3×3 checkerboard ($\alpha = 1.0$) (d) 3×3 random ($\alpha = 0.2$) (e) 3×3 random ($\alpha = 1.0$)

(f) 2×2 checkerboard ($\alpha = 0.2$) (g) 2×2 checkerboard ($\alpha = 1.0$) (h) Blended ($\alpha = 0.05$) (i) Blended ($\alpha = 0.2$) (j) Label consistent

Figure 2: Examples of poisoned images in bird class. (a) is the original image. (b)–(j) are the poisoned ones with different types of triggers.

Experimental Settings. To explore the intersection between the perturbation budget and the trigger magnitude, we use varying perturbation budget ϵ of AT from $2/255$ to $16/255$, and varying transparency α from 0.2 to 1.0. Besides, we train models in common settings: The normally and adversarially trained ResNet-18 (He et al., 2016) models are obtained using an SGD optimizer with the momentum 0.9, the weight decay 5×10^{-4}, and the initial learning rate 0.1 which is divided by 10 at the 60-th and 90-th epochs. To craft adversarial examples, we apply PGD attack, where the step size is $2/255$ and the number of steps is 10. To inject backdoor behaviors, we apply the clean label setting, where only data belonging to the target class are poisoned, following Weng et al. (2020) and the trigger pattern is a 3×3 checkerboard as shown in Figure 9. The target label is class 2 (bird) and the poisoning rate is 5% (250/5000). For simplicity, we refer to this as the basic setting.

3.1 DOES AT REALLY IMPLY BACKDOOR VULNERABILITY?

Figure 3(a) visualizes how the attack success rate (ASR) varies with respect to different perturbation budget ϵ in the basic setting, where ASR is the ratio of triggered samples that are misclassified as the target label, indicating the backdoor vulnerability. The corresponding clean accuracies can be found in Appendix B.

Small Perturbation Budget Strengthens Backdoor Vulnerability. In Figure 3(a), for the fixed trigger transparency $\alpha = 0.6$ (the blue line), as long as the perturbation budget of AT is not too large

[1]There are also some scenarios where the adversary can control the whole training process (Li et al., 2020a; Pang et al., 2020), which only occurs in outsourcing training in untrustworthy third parties or pretrained models from unknown sources.

($0 \leq \epsilon \leq 8/255$), the ASR always increases when we adversarially train models on the poisoned training set. This is consistent with the findings in previous research (Weng et al., 2020), that is, we obtain adversarial robustness at the cost of the drop in backdoor robustness. It is interesting that within this range with small budgets ($\leq 8/255$), the ASR continues to increase when we use slightly larger budgets. For an extremely small transparency $\alpha = 0.2$ (the red line), we still can observe similar trends within a smaller range ($\epsilon < 0.05/255$) as shown in Figure 4(b). Overall, we can always find small budgets of AT strengthens backdoor vulnerability.

Large Perturbation Budget Mitigates Backdoor Vulnerability. Here, we enlarge the perturbation budgets. When the perturbation budget exceeds a certain threshold ($\epsilon = 8/255$ for $\alpha = 0.6$), AT significantly decreases ASR and improves the backdoor robustness in Figure 3(a). This is the neglected part by previous research (Weng et al., 2020). In addition, the larger the budget is, the more robust (against backdoor attacks) the model is. For an extremely large transparency $\alpha = 0.8$ or 1.0, we are still able to decrease the ASR to a low value with a much larger perturbation budget ($\epsilon = 24/255$) as shown in Figure 4(a).

An Intuitive Understanding on the Intersection. As indicated in Ilyas et al. (2019), the adversarially trained model learns robust features (predictive and robust) and ignores the non-robust features (predictive yet brittle). Since the backdoor trigger is usually difficult to modify by small adversarial perturbations, it can be regarded as a robust feature. As a result, AT helps model learn the backdoor-related feature. Besides, the normally trained model relies on both robust and non-robust features, while the adversarially trained model only relies on robust features, letting the backdoor-related robust feature contributes more in AT. However, as the perturbation budget exceeds some threshold and is able to modify the trigger pattern, the backdoor trigger becomes a non-robust feature instead. Thus, AT hinders the model from learning the backdoor-related feature.

3.2 The Consistency of the Experimental Results

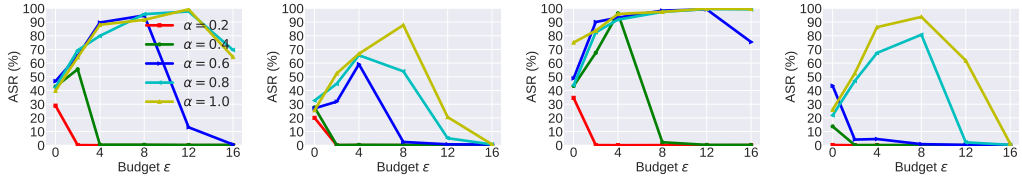
To verify whether the phenomenon above is common, we conduct experiments across different settings, including poisoning rates (see Figures 3(b)–3(c)), poisoning types (see Figure 3(f)), trigger patterns (see Figure 3(d)), trigger sizes (see Figure 4(b)), architectures (see Figure 3(g)) and target classes (see Figure 3(h)). Besides, we also explore the interactions between AT and other advanced backdoor attacks including the blended backdoor attack (Chen et al., 2017) and the label-consistent attack (Turner et al., 2019) in Figures 4(c)–4(d). Specifically, the blended backdoor attack blends benign data with a predefined Gaussian noise pattern, while the label consistent attack leverages the adversarial perturbations to improve the attack efficiency. We find: (i) AT is able to strengthen the backdoor vulnerability when the perturbation budget is small; (ii) as long as the perturbation budget exceeds a specific threshold, AT always prevents models from learning the backdoor behaviors and consistently improves the backdoor robustness, which is neglected by previous research (Weng et al., 2020). More details about the experimental settings can be found in Appendix A.

4 AT-based Detection for Poisoned Data

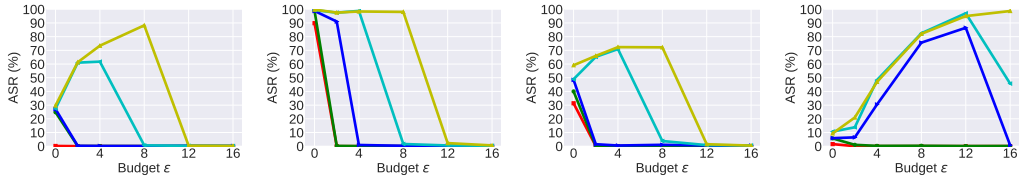
As mentioned above, AT is only able to prevent models from learning backdoor behaviors when the perturbation budget is large enough. Unfortunately, AT strengthens backdoor vulnerability with small budgets instead, which brings severe security threats to practical scenarios. In this section, we try to propose an efficient defense strategy in the latter situation.

Differently from ST, the inner-loop optimization of AT generates adversarial data by maximizing the adversarial loss with PGD iterations. Thus the number of PGD steps is crucial for both the training samples and final models. One category of works has been devoted to investigating the effect of PGD steps on the model's convergence (Wang et al., 2019b) and robustness (Zhang et al., 2020; 2021), while another work utilized the PGD steps to determine whether a sample is difficult or easy to learn for current models (Zhang et al., 2021; Zhu et al., 2021). Inspired by this, we investigate the number of steps required to successfully attack poisoned samples (*i.e.*, fooling DNN to predict different from the annotation) during AT.

We apply the minimum number of steps for a successful attack to investigate the geometric property of training data: Given a model (that might be a checkpoint during training) and a sample, we count the **minimum number of steps** required to make DNN predict different from its annotation without

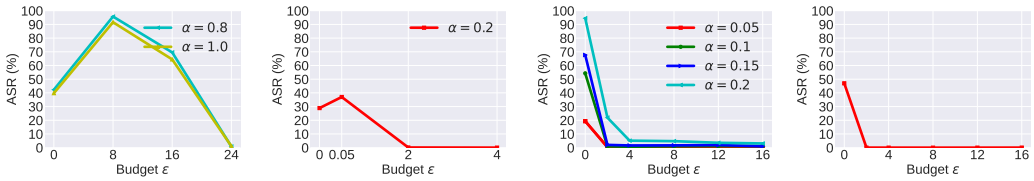


(a) Basic settings (b) Poison rate = 3% (c) Poison rate = 8% (d) Random trigger



(e) 2×2 trigger (f) Dirty labels (g) VGG16 (h) Ship class

Figure 3: Attack success rates of adversarially and normally trained models. $\epsilon = 0$ means standard training. (a) is the results of basic settings. (b) and (c) are the results with varying poison rates. (d) and (e) are the results with different triggers. (f) is the result with poisoning samples not restricted to the target class. (g) is the result when replacing the ResNet-18 with VGG16 network (Simonyan & Zisserman, 2014). (h) is the result when replacing the target bird class with ship class.



(a) $\epsilon = 24/255$ (b) $\epsilon = 0.05/255$ (c) Blended (d) Label-consistent

Figure 4: (a) and (b) are the comparisons with larger or smaller budget ϵ in basic settings. (c) and (d) are the results of blended attacks and label-consistent attacks.

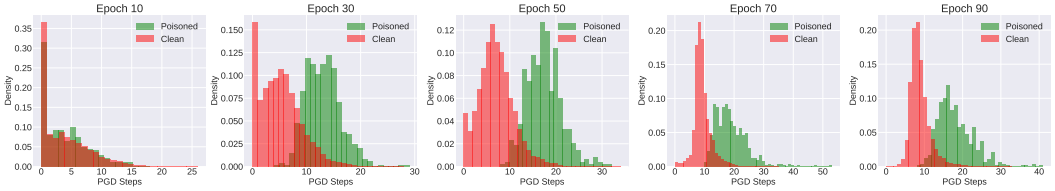
any l_p norm restriction, *i.e.*, $\|x - x'\|_p \leq \epsilon$[2], since we want every sample from training data to be attacked successfully after several steps. The detailed description is in Algorithm 1.

Algorithm 1 The Minimum Number of Steps

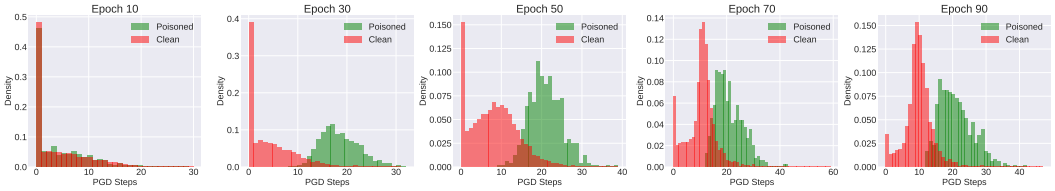
Input:
training data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, evaluation step size β, loss function $\mathcal{L}$, classifier f_θ;
Output:
PGD steps $\{\tau_i\}_{i=1}^n$;
for $(x_i, y_i) \in \mathcal{D}$ **do**
  $\tau_i \leftarrow 0$;
  while $f(x_i) = y_i$ **do**
    $x_i \leftarrow \Pi_{[0,1]^d}\left[x_i + \beta \cdot \text{sign}(\bigtriangledown_x \mathcal{L}(f_\theta(x_i), y_i))\right]$;
    $\tau_i \leftarrow \tau_i + 1$;
  end while
end for
return $\{\tau_i\}_i^n$;

Experimental Settings for Minimum Number of Steps. We adopt the basic attack in Eq. (3) with $\alpha = 1.0$ and the 3×3 checkerboard trigger. We poison 10% of samples (500 images) belonging

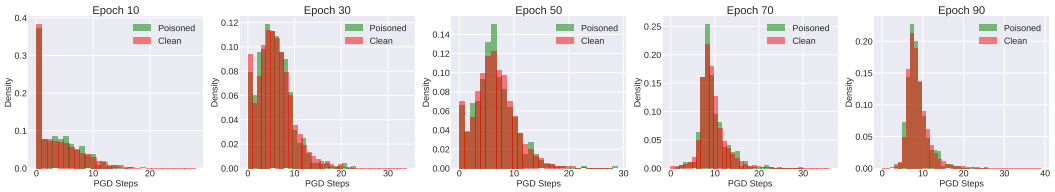
[2]We still project the generated image into the valid pixel space $[0, 1]^d$.



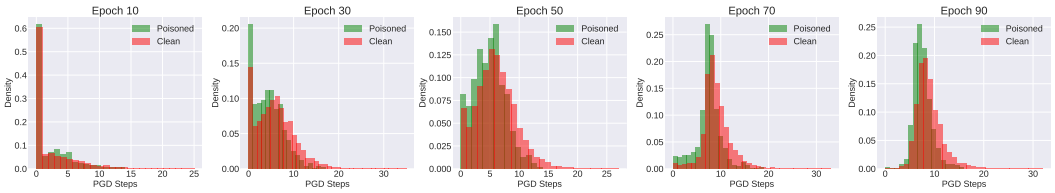
(a) $\epsilon = 8/255, \alpha = 1.0$ (backdoor robustness $\downarrow$)



(b) $\epsilon = 12/255, \alpha = 1.0$ (backdoor robustness $\downarrow$)



(c) $\epsilon = 8/255, \alpha = 0.2$ (backdoor robustness $\uparrow$)



(d) $\epsilon = 8/255, \alpha = 0.2$, blended attacks (backdoor robustness $\uparrow$)

Figure 5: The density statistics of PGD steps for clean and poisoned samples.

to the class "bird" (class 2) with the trigger. Other settings for adversarial training are the same as the basic setting in Section 3. With the fixed step size $2/255$, we record and visualize the minimum number of steps to successfully attack the input sample as shown in Figures 5(a) and 5(b). We find that when AT ($\epsilon = 8/255, 12/255$) strengthens the memorization of distinct triggers ($\alpha = 1.0$), the minimum number of steps to attack poisoned samples are significantly larger than those to attack clean samples. This is because, in our opinion, the adversarially trained model memorizes trigger pattern better, which brings more difficulties to craft an adversarial examples based on them to fool the model. Moreover, after several epochs (*e.g.*, 10 epochs), this phenomenon exists across the entire training process. Meanwhile, we also find that when AT mitigates backdoor vulnerability, the minimum number of PGD steps is unable to identifying poisoned samples (see Figures 5(c) and 5(d)) anymore. Fortunately, adversarial training can mitigate indistinct backdoor triggers as shown in Section 3. More experiments about the minimum number of steps can be found in Appendix D.

PGD Defense. Here, we propose a detection strategy termed PGD defense. As discussed above, samples required more steps are believed to be poisoned and should be filtered out before fed into the final training. Specifically, we first adversarially train a model based on untrusted samples until a predefined epoch (T epoch) as the warm-up phase. Then we calculate and record the minimum number of steps to attack each training sample with the current model. Following previous backdoor detection works (Tran et al., 2018; Chen et al., 2018), we remove $1.5p \cdot n$ (p is the poison rate and n is the number of total samples) samples with the largest PGD steps. The whole procedure of PGD defense is summarized in Algorithm 2.

To demonstrate the effectiveness, we compare the proposed PGD defense with other detection methods: PCA defense (Tran et al., 2018) and clustering defense (Chen et al., 2018). In PGD defense, we use $\epsilon = 8/255$ in AT in the warm-up phase with $T = 20$ and $\beta = 2/255$ to evaluate PGD steps. All the baseline methods are executed based on the official open-source implementations with default parameters. In experiments, we varied the poison rate and transparency parameter to observe the detection results under different conditions. Table 1 shows the detection results in various settings, from which we find that PGD defense achieves comparable performance to state-of-the-art detection methods.

Algorithm 2 PGD defense.

Input:
training data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, perturbation size ϵ, loss function $\mathcal{L}$, batch size B, evaluation step size β, poison rate p, warm-up epoch T, classifier f_θ;
Output:
Purified training data $\mathcal{D}'$;
$\theta \leftarrow \theta_0, t \leftarrow 0$;
while $t < T$ **do**
  Sample a mini-batch data $\{x_i, y_i\}_{i=1}^B$ from $\mathcal{D}$;
  Compute adversarial data $\{\tilde{x_i}, y_i\}_{i=1}^B$ according to Eq. (2);
  Update $\theta \leftarrow \theta - \frac{1}{B}\sum_{i=1}^B \bigtriangledown_\theta \mathcal{L}(f_\theta(\tilde{x}_i), y_i)$;
  $t \leftarrow t + 1$;
end while
Compute the PGD steps $\{\tau_i\}_i^n$ with f_θ and β according to Algorithm 1;
$\mathcal{D}' \leftarrow \mathcal{D} \setminus \{1.5p \cdot n$ samples with greatest τ_i values$\}$;
return $\mathcal{D}'$;

Table 1: The comparison of detection results with other baseline detection methods. "732/750" means 732 poisoned samples are removed among total 750 poisoned samples.

Poison rate	α	PCA defense	Clustering defense	PGD defense
20%	1.0	1000/1000	1000/1000	1000/1000
15%	1.0	750/750	750/750	732/750
10%	1.0	500/500	500/500	500/500
5%	1.0	250/250	250/250	217/250
20%	0.8	1000/1000	1000/1000	1000/1000
15%	0.8	749/750	748/750	747/750
10%	0.8	500/500	500/500	500/500
5%	0.8	132/250	85/250	170/250

The Pipeline of Adversarial Training against Backdoor Attacks. To tackle both the adversarial and backdoor risks, we propose a two-phase pipeline of adversarial training against backdoor attacks regardless of the trigger magnitude. In the first phase, we first use AT to acquire a detection model and then apply PGD defense to remove the poisoned samples. In the second phase, we retrain the purified training set with AT to achieve both high adversarial and backdoor robustness. We explain the rationality of such design: if AT strengthens backdoor attacks, the poisoned samples are effectively removed in the first phase and if not, adversarial retraining will mitigate backdoor vulnerability in the second phase although a small portion of benign samples are wrongly removed in the first phase.

Comparisons with Adversarial Training and Standard Training. The evaluation criteria include three aspects: the clean accuracy (available for normal usage), adversarial accuracy (the ability to defend adversarial attack) and attack success rate (the ability to defend backdoor attack). We conducted experiments using basic attacks (Eq. (3)) with $\alpha = 1.0$ (distinct triggers) an $\alpha = 0.2$ (indistinct triggers) to ensure the practicality of the proposed pipeline when we know nothing about the backdoor attacks. The first and second phase in the pipeline both adopt $\epsilon = 8/255$ adversarial training. The adversarial accuracy is evaluated with $\epsilon = 8/255$ PGD attacks. The results are summarized in Figure 7. We observe that although AT obtains higher adversarial robustness compared with ST, AT strengthens backdoor vulnerability when the trigger magnitude is large ($\alpha = 1.0$). For our

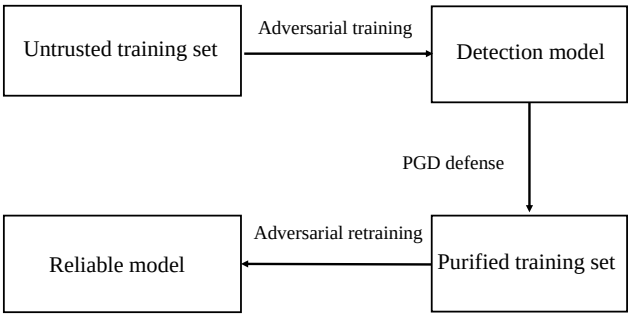
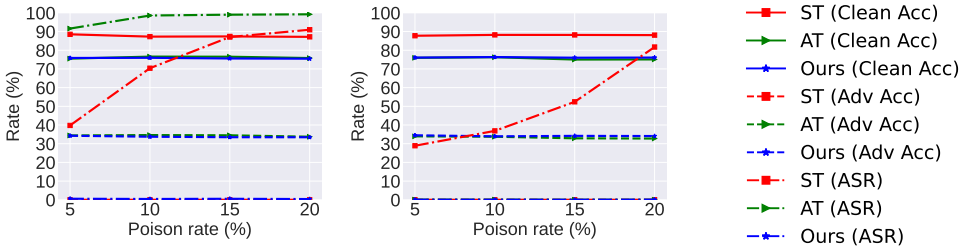


Figure 6: The pipeline of adversarial training against backdoor attacks.



(a) $\alpha = 1.0$ (b) $\alpha = 0.2$

Figure 7: Experimental results of standard training, adversarial training and our pipeline.

pipeline, we find that when the trigger magnitude is large ($\alpha = 1.0$), the first phase of the pipeline can effectively remove the poisoned samples and when the trigger magnitude is small ($\alpha = 0.2$), the second phase of the pipeline can completely mitigate backdoor behaviors although the first phase is ineffective. Therefore our pipeline achieves high adversarial and backdoor robustness at the same time. More results can be found in Appendix C.

5 Conclusion

In this work, we delved into the interactions between adversarial training and backdoor attacks. We first challenged the trade-off between adversarial and backdoor robustness and conducted comprehensive experiments to investigate the conditions under which the trade-off occurs. We found that adversarial training indeed mitigates backdoor attacks as long as the perturbation budget suppresses the trigger distinctness. Then we focused on the circumstances that adversarial training strengthens backdoor attacks and leverage the number of PGD steps for successful attacks to detect poisoned samples. Finally, we proposed a general pipeline of adversarial training against backdoor attacks regardless of the trigger pattern.

ETHICS STATEMENT

We have known that DNNs are usually built by harvesting data from unverified sources and thus the adversary may perform poisoning based attacks. Our work considers one stream of extensively studied attacks named as backdoor attacks and propose a general adversarial training based pipeline against backdoor attacks. Our method will benefit both backdoor and adversarial robustness and we hence do not find any potential harmful insights for the whole society. Also, the datasets used in our experiments are public and do not involve privacy or copyright concerns.

REPRODUCIBILITY STATEMENT

To ensure the reproducibility of our experiments, we have uploaded the source code as a zip file. Almost all the experiments can be reproduced with the code except the baseline backdoor detection methods which can be executed with the authors' open source implementations. Besides, we also include the running commands and required packages.

REFERENCES

Jean-Baptiste Alayrac, Jonathan Uesato, Po-Sen Huang, Alhussein Fawzi, Robert Stanforth, and Pushmeet Kohli. Are labels required for improving adversarial robustness? In *NeurIPS*, 2019.

Farman Ali, Shaker El-Sappagh, SM Riazul Islam, Daehan Kwak, Amjad Ali, Muhammad Imran, and Kyung-Sup Kwak. A smart healthcare monitoring system for heart disease prediction based on ensemble deep learning and feature fusion. *Information Fusion*, 63:208–222, 2020.

Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C Duchi, and Percy S Liang. Unlabeled data improves adversarial robustness. In *NeurIPS*, 2019.

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 preprint arXiv:1811.03728*, 2018.

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

Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

Xinyun Chen, Wenxiao Wang, Chris Bender, Yiming Ding, Ruoxi Jia, Bo Li, and Dawn Song. Refit: a unified watermark removal framework for deep learning systems with limited data. In *AsiaCCS*, 2021.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

Ilias Diakonikolas, Gautam Kamath, Daniel Kane, Jerry Li, Jacob Steinhardt, and Alistair Stewart. Sever: A robust meta-algorithm for stochastic optimization. In *ICML*, 2019.

Shaohua Ding, Yulong Tian, Fengyuan Xu, Qun Li, and Sheng Zhong. Trojan attack on deep generative models in autonomous driving. In *International Conference on Security and Privacy in Communication Systems*, pp. 299–318. Springer, 2019.

Micah Goldblum, Dimitris Tsipras, Chulin Xie, Xinyun Chen, Avi Schwarzschild, Dawn Song, Aleksander Madry, Bo Li, and Tom Goldstein. Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses. *arXiv preprint arXiv:2012.10544*, 2020.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

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. *arXiv preprint arXiv:1908.01763*, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Hongsheng Hu, Zoran Salcic, Gillian Dobbie, and Xuyun Zhang. Membership inference attacks on machine learning: A survey. *arXiv preprint arXiv:2103.07853*, 2021.

Andrew Ilyas, Shibani Santurkar, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *NeurIPS*, 2019.

Minseon Kim, Jihoon Tack, and Sung Ju Hwang. Adversarial self-supervised contrastive learning. *arXiv preprint arXiv:2006.07589*, 2020.

Klim Kireev, Maksym Andriushchenko, and Nicolas Flammarion. On the effectiveness of adversarial training against common corruptions. *arXiv preprint arXiv:2103.02325*, 2021.

Yiming Li, Baoyuan Wu, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor learning: A survey. *arXiv preprint arXiv:2007.08745*, 2020a.

Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu. Backdoor attack with sample-specific triggers. *arXiv preprint arXiv:2012.03816*, 2020b.

Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *International Symposium on Research in Attacks, Intrusions, and Defenses*, 2018.

Xuankai Liu, Fengting Li, Bihan Wen, and Qi Li. Removing backdoor-based watermarks in neural networks with limited data. In *ICPR*, 2021.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.

Tuan Anh Nguyen and Anh Tran. Input-aware dynamic backdoor attack. In *NeurIPS*, 2020.

Rui Ning, Jiang Li, Chunsheng Xin, and Hongyi Wu. Invisible poison: A blackbox clean label backdoor attack to deep neural networks. In *INFOCOM*, 2021.

Ren Pang, Zheng Zhang, Xiangshan Gao, Zhaohan Xi, Shouling Ji, Peng Cheng, and Ting Wang. Trojanzoo: Everything you ever wanted to know about neural backdoors (but were afraid to ask). *arXiv preprint arXiv:2012.09302*, 2020.

Nicolas Papernot, Patrick McDaniel, Arunesh Sinha, and Michael P Wellman. Sok: Security and privacy in machine learning. In *EuroS&P*, 2018.

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 *ECCV*, 2020.

Nikolaos Pitropakis, Emmanouil Panaousis, Thanassis Giannetsos, Eleftherios Anastasiadis, and George Loukas. A taxonomy and survey of attacks against machine learning. *Computer Science Review*, 2019.

Chongli Qin, James Martens, Sven Gowal, Dilip Krishnan, Krishnamurthy Dvijotham, Alhussein Fawzi, Soham De, Robert Stanforth, and Pushmeet Kohli. Adversarial robustness through local linearization. In *NeurIPS*, 2019.

Aniruddha Saha, Akshayvarun Subramanya, and Hamed Pirsiavash. Hidden trigger backdoor attacks. In *AAAI*, 2020.

Hadi Salman, Andrew Ilyas, Logan Engstrom, Ashish Kapoor, and Aleksander Madry. Do adversarially robust imagenet models transfer better? In *NeurIPS*, 2020.

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. In *NeurIPS*, 2018.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 2016.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Jacob Steinhardt, Pang Wei Koh, and Percy Liang. Certified defenses for data poisoning attacks. In *NeurIPS*, 2017.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *ICLR*, 2014.

Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. In *NeurIPS*, 2018.

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

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 S&P*, 2019a.

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 *ECCV*, 2020.

Yisen Wang, Xingjun Ma, James Bailey, Jinfeng Yi, Bowen Zhou, and Quanquan Gu. On the convergence and robustness of adversarial training. In *ICML*, 2019b.

Cheng-Hsin Weng, Yan-Ting Lee, and Shan-Hung Brandon Wu. On the trade-off between adversarial and backdoor robustness. 2020.

Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. In *NeurIPS*, 2020.

Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan L Yuille, and Quoc V Le. Adversarial examples improve image recognition. In *CVPR*, 2020.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *ICML*, 2019.

Jingfeng Zhang, Xilie Xu, Bo Han, Gang Niu, Lizhen Cui, Masashi Sugiyama, and Mohan Kankanhalli. Attacks which do not kill training make adversarial learning stronger. In *ICML*, 2020.

Jingfeng Zhang, Jianing Zhu, Gang Niu, Bo Han, Masashi Sugiyama, and Mohan Kankanhalli. Geometry-aware instance-reweighted adversarial training. In *ICLR*, 2021.

Chen Zhu, W Ronny Huang, Hengduo Li, Gavin Taylor, Christoph Studer, and Tom Goldstein. Transferable clean-label poisoning attacks on deep neural nets. In *ICML*, 2019.

Jianing Zhu, Jingfeng Zhang, Bo Han, Tongliang Liu, Gang Niu, Hongxia Yang, Mohan Kankanhalli, and Masashi Sugiyama. Understanding the interaction of adversarial training with noisy labels. *arXiv preprint arXiv:2102.03482*, 2021.

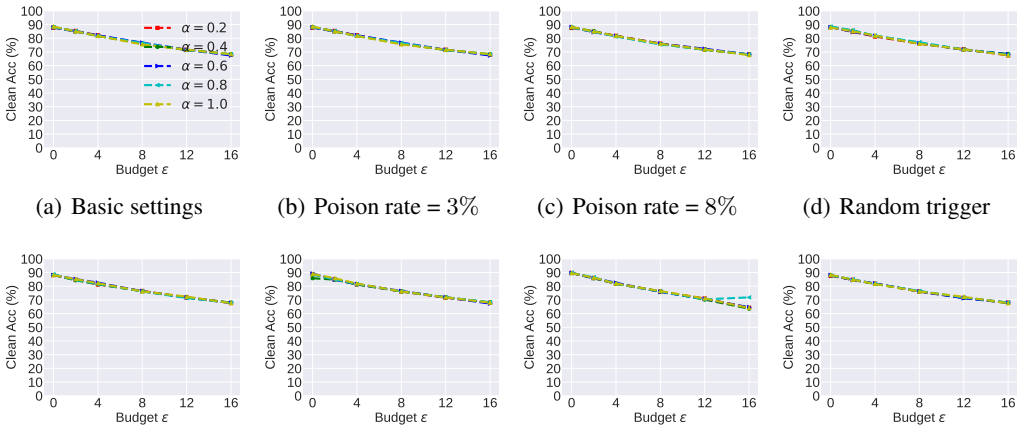
A More Implementation Details on Backdoor Attacks

Experiments on blended attacks. We generate a random trigger pattern $r \in [0, 1]^d$ with the same size as the original image and then blend both with a mixing weight α. We train a ResNet-18 model with poisoning 5% samples in bird class and vary α from 0.05 to 0.2 . Other training details such as epochs and learning rates are the same as basic settings.

Experiments on label-consistent attacks. We first train a robust ResNet-18 model with AT. We set the perturbation bound $\epsilon = 8/255$ and step size $2/255$. The total epochs is 120 and the initial learning rate is 0.1 which is divide by 10 in epoch 60 and 90. Then we add the adversarial perturbations generated by the robust model to the original images with PGD attacks. To execute stealthier attacks, we perturb the original pixel with the backdoor trigger amplitude $16/255$ (Turner et al., 2019). Other details such as poison rates and the final training procedures are the same as basic settings.

B Clean accuracies of the victim models

We visualize the clean accuracies involved in our experiments in Figure 8 (corresponds to Figure 3).

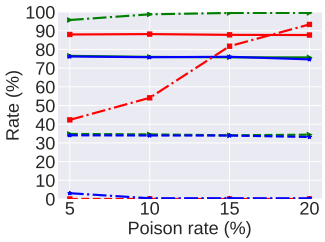


(a) Basic settings (b) Poison rate = 3% (c) Poison rate = 8% (d) Random trigger

(e) 2×2 trigger (f) Dirty labels (g) VGG16 (h) Ship class

Figure 8: Clean accuracies of adversarially and normally trained models.

C More experiments about the proposed pipeline

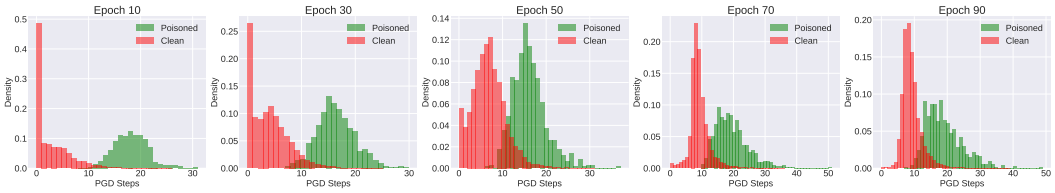


(a) $\alpha = 0.8$

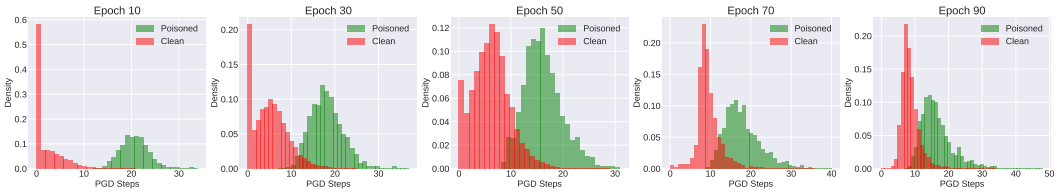
Figure 9: More experiments with our proposed pipeline.

D MORE DENSITY STATISTICS OF PGD STEPS WITH ADVERSARIAL TRAINING.

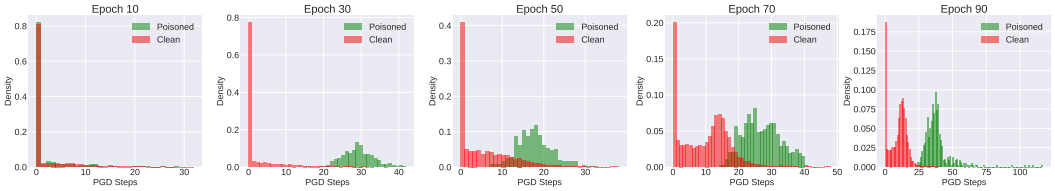
We also compute the PGD steps in more cases and the statistics of clean and poisoned samples are always distinguishable whenever the backdoor attacks are strengthened (see Figure 10).



(a) $\epsilon = 8/255, \alpha = 1.0$, poison rate=15% (backdoor robustness ↓)



(b) $\epsilon = 8/255, \alpha = 1.0$, poison rate=20%, (backdoor robustness ↓)



(c) $\epsilon = 16/255, \alpha = 1.0$, poison rate=10% (backdoor robustness ↓)

Figure 10: The density statistics of PGD steps for clean and poisoned samples.