Beating Backdoor Attack at Its Own Game

Min Liu[1], Alberto Sangiovanni-Vincentelli[2], Xiangyu Yue[3]
[1]Carnegie Mellon University, [2]UC Berkeley, [3]The Chinese University of Hong Kong
minliu2@cs.cmu.edu, alberto@berkeley.edu, xyyue@ie.cuhk.edu.hk

Abstract

Deep neural networks (DNNs) are vulnerable to backdoor attack, which does not affect the network's performance on clean data but would manipulate the network behavior once a trigger pattern is added. Existing defense methods have greatly reduced attack success rate, but their prediction accuracy on clean data still lags behind a clean model by a large margin. Inspired by the stealthiness and effectiveness of backdoor attack, we propose a simple but highly effective defense framework which injects non-adversarial backdoors targeting poisoned samples. Following the general steps in backdoor attack, we detect a small set of suspected samples and then apply a poisoning strategy to them. The non-adversarial backdoor, once triggered, suppresses the attacker's backdoor on poisoned data, but has limited influence on clean data. The defense can be carried out during data preprocessing, without any modification to the standard end-to-end training pipeline. We conduct extensive experiments on multiple benchmarks with different architectures and representative attacks. Results demonstrate that our method achieves state-of-the-art defense effectiveness with by far the lowest performance drop on clean data. Considering the surprising defense ability displayed by our framework, we call for more attention to utilizing backdoor for backdoor defense. Code is available at https://github.com/damianliumin/non-adversarial_backdoor.

1. Introduction

In recent years, deep neural networks (DNNs) have achieved impressive performance across tasks, such as object detection [36, 33], speech recognition [46, 2] and machine translation [37, 41]. With the increasing usage of DNNs, security of neural networks has attracted a lot of attention. Studies have shown that DNNs are especially vulnerable to backdoor attack [43], a variant of data poisoning which fools the model to establish a false correlation between inserted patterns and target classes. Specifically, the adversary injects a trigger pattern to a small proportion of the training data. A network trained on the poisoned data has normal behavior on benign data, but deviates from its expected output when the trigger pattern is implanted.

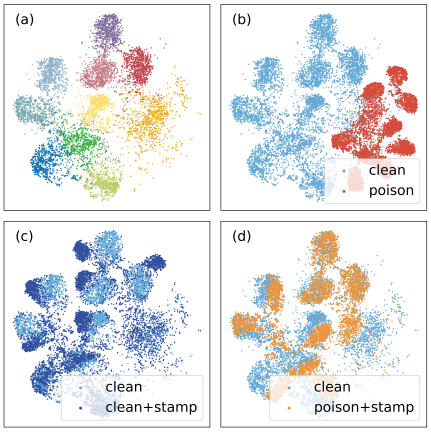


Figure 1: Representations under the effect of adversarial backdoor (AB) and non-adversarial backdoor (NAB), which are injected by attackers and defenders respectively. "Stamp" is the trigger pattern for NAB. (a) Clean samples are not influenced by backdoor. (b) AB changes model behavior on poisoned samples. (c) NAB is not triggered on clean samples. (d) NAB suppresses the effectiveness of AB on poisoned samples.

To ensure the security of DNN systems, a lot of novel defense methods have been proposed in the past few years. Most of the defense methods try to either 1) avoid learning the backdoor during training or 2) erase it from a poisoned model at the end. Following idea 1), some studies detect and filter poisoned samples [38, 6]. Since a small number of poisoned samples slipping from detection can lead to a successful attack, simply filtering the potentially poisoned samples is not enough in most cases. A more realistic way is to adopt data separation as an intermediate procedure [23, 14]. Some other works pre-process the input to depress the effectiveness of injected patterns [31, 13]. However, these methods have limited effects under the increasingly diverse attacking strategies. Another line of work follows idea 2) [24, 44]. Despite promising defense effectiveness,

erasing-based methods suffer from performance drop due to the additional erasing stage. Performance on clean data still lags behind a clean model by a large margin. Reducing the performance gap on clean data while maintaining satisfying defense effectiveness remains a challenging problem.

Under backdoor attack, representations of poisoned samples are dominated by the trigger pattern as shown in Fig. 1. Therefore, injecting the pattern can force a poisoned model to behave in a way expected by the attacker. Considering the effectiveness of such strategies, a natural question is whether backdoor can be utilized for defense purpose, that is to say *beating backdoor attack at its own game*. To be more specific, a model might misbehave when only the trigger pattern is exposed, but the mishehavior should be suppressed once a benign pattern, which is called a *stamp* in this paper, is injected to the poisoned sample. There are three advantages behind this idea. *First*, the defender only needs a small set of poisoned training samples to inject a backdoor, which is a much easier requirement than filtering all the poisoned data. *Second*, a backdoor targeting poisoned data, ideally, will not influence the model performance on clean data. *Finally*, the backdoor can be injected during data pre-processing, without any modification to the standard end-to-end training pipeline.

In this work, we propose a novel defense framework, *Non-adversarial Backdoor (NAB)*, which suppresses backdoor attack by injecting a backdoor targeting poisoned samples. Specifically, we first detect a small set of suspected samples using existing methods such as [23, 14, 11]. Then we process these samples with a poisoning strategy, which consists of a stamping and a relabeling function. A pseudo label is generated for each detected sample and we stamp the samples with inconsistent orginal and pseudo labels. In this way, we insert a non-adversarial backdoor which, once triggered, is expected to change model behaviors on poisoned data. Furthermore, NAB can be augmented with an efficient test data filtering technique by comparing the predictions with or without the stamp, ensuring the performance on poisoned data. We instantiated the NAB framework and conducted experiments on CIFAR-10 [16] and tiny-ImageNet [17] over several representative backdoor attacks. Experiment results show that the method achieves state-of-the-art performance in both clean accuracy and defense effectiveness. Extensive analyses demonstrate how NAB takes effect under different scenarios.

Our main contributions can be summarized as follows:

- We propose the idea of backdooring poisoned samples to suppress backdoor attack. To the best of our knowledge, our work is the first to utilize non-adversarial backdoor in backdoor defense.
- We transform the idea into a simple, flexible and effective defense framework, which can be easily augmented with a test data filtering technique.
- Extensive experiments are conducted and our method achieves state-of-the-art defense effectiveness with by far the lowest performance drop on clean data.

2. Related Work

Backdoor Attack. Backdoor attack is a type of attack involved in the training of DNNs, with the interesting property that the model works well on clean data but generates unexpected outputs once the attack is triggered. A main track of the attacks focuses on poisoning training data in an increasingly stealthier and more effective way by developing novel trigger patterns [15]. Attack methods for visual models, the mainstream of backdoor attack research, can be divided according to the visibility of patterns. Visible attacks inject human perceptible patterns like a single pixel [39], an explicit patch [10, 26], sample-specific patterns [30], or more complex and indistinguishable patterns like blending random noise [5] and sinusoidal strips [3]. Invisible attacks [50, 40, 34, 18, 29, 22] are even more stealthy to human observers. Backdoor attacks can also be categorized into dirty-label attacks [10, 26, 30] and clean-label attacks [3, 40]. Clean-label attacks are more difficult to detect since there lacks an obvious mismatch between the images and labels. We also notice some non-poisoning based methods which induce backdoor by modifying other training settings [19, 20] or the model weights [7, 9, 32].

Backdoor Defense. Existing backdoor defense methods aim to avoid learning the backdoor during training or erase the backdoor at the end. To avoid injection of the backdoor, various techniques detecting poisoned data have been proposed [39, 4, 38, 6, 11]. These methods alone cannot achieve successful defense when a fraction of poisoned samples escape from the detection. Instead of simply filtering all the poisoned samples, a more practical idea is to adopt data separation as an intermediate procedure. Some other works attempt to bypass the backdoor by pre-processing the input before passing it into the model [31, 8, 13], but these methods typically have limited effects over the increasingly various attacks. Meanwhile, erasing methods try to mitigate the effects of backdoor after the model gets attacked [23, 48, 24, 44]. [23] reduced attack success rate to a negligible level under several attacks, but the prediction accuracy on clean data still lags behind a well-trained clean model by a large margin. Our method adopts a data separation stage as in [23, 14]. Nevertheless, the core idea, injecting a backdoor for defense purpose, is similar to none of the previous defense methods.

Non-Adversarial Backdoor. Non-adversarial applications of backdoor has been proposed before, including watermark-based authentication [1], protection of open-sourced datasets [25] and neural networks interpretability [49]. [35] also injected a backdoor to hide weaknesses in a model under adversarial attack [28]. However, our work is

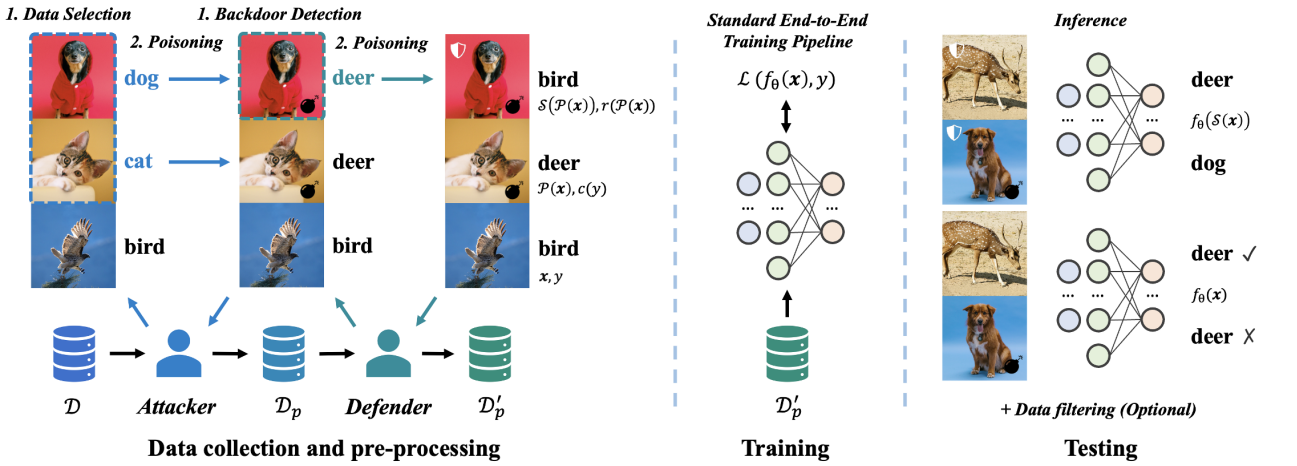


Figure 2: Overview of the proposed framework. The attacker injects an adversarial backdoor by selecting and poisoning a set of clean samples. After obtaining the dataset, the defender detect and poison a set of suspected samples to inject the non-adversarial backdoor. Both attack and defense take place in the standard end-to-end training pipeline. In the testing stage, we stamp each input to keep the non-adversarial backdoor triggered. We can also adopt a test data filtering technique by comparing the predictions with or without the stamp. Samples with inconsistent predictions are identified as poisoned.

the first attempt to utilize backdoor in defense against backdoor attack.

3. Method

3.1. Preliminary

Threat Model. In this paper, we assume that the attacker has full control over the data source, is capable of arbitrarily changing the images and relabeling them with target classes, but does not have access to the model and training process. The defender has control over the model, training process, and data once obtained from the data source, but does not know the proportion and distribution of the poisoned samples, target classes, and attacking strategies. Given some partially poisoned data, the defender aims to train a model that preserves accuracy on clean data and avoids predicting the target class on poisoned data.

Backdoor Attack. The attacker first obtains a clean training set $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$, where $\boldsymbol{x}_i \in \mathcal{X}$ is an image and $y_i \in \mathcal{Y}$ is the corresponding label. The poisoning strategy consists of two parts: $\mathcal{P} : \mathcal{X} \to \mathcal{X}$ applies a trigger pattern to the image and $c : \mathcal{Y} \to \mathcal{Y}$ replaces the label with target label. The attacker selects a subset of clean data and generates a set of malign samples $\mathcal{D}_m = \{(\mathcal{P}(\boldsymbol{x}), c(y))\}$ accordingly using the poisoning strategy, where $\lambda = |\mathcal{D}_m|/|\mathcal{D}|$ is the poisoning rate. Merging $\mathcal{D}_m$ with the remaining clean training data, the attacker generates a poisoned dataset $\mathcal{D}_p$ and releases it to potential victims.

The empirical error on a poisoned dataset can be decomposed into a *clean loss* and an *attack loss* [23]:

$$\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[\ell(f_\theta(\boldsymbol{x}), y)] + \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_m}[\ell(f_\theta(\boldsymbol{x}), y)] \quad (1)$$

where $\ell(\cdot)$ is the loss function and $f_\theta : \mathcal{X} \to \mathcal{Y}$ is the neural network. Minimizing the first term above encourages the model to learn the image classification task while the second one forces the learning of a correlation between the trigger pattern and target class. Both tasks can be learned well due to the excessive learning ability of neural networks [21], making backdoor attack effective and hard to detect.

3.2. Non-Adversarial Backdoor

The success of backdoor attack leads us to think about the feasibility of utilizing backdoor for defense purpose. An attacker wants the model to classify a benign sample $\boldsymbol{x}$ to the target class. In the same way, a defender wants the model to classify a poisoned sample $\mathcal{P}(\boldsymbol{x})$ to any but the target class. The similarity between the objectives makes it a natural idea to apply backdoor in both attack and defense settings, while the latter was rarely explored.

Based on the idea above, we propose a defense framework *Non-Adversarial Backdoor* (NAB). In this section, we assume that a small set of suspected samples $\mathcal{D}'_s \subset \mathcal{D}_p$ and a poisoning strategy are available. The defender's poisoning strategy also has two components: 1) $\mathcal{S} : \mathcal{X} \to \mathcal{X}$ applies a trigger pattern, which is called a *stamp* to tell from the adversarial trigger pattern, and 2) $r : \mathcal{X} \to \mathcal{Y}$ generates a pseudo label conditioned on the image. Details of backdoor detection and poisoning strategies are discussed in Sec. 3.3. We then generate a set of stamped samples $\mathcal{D}'_m = \{(\mathcal{S}(\boldsymbol{x}), r(\boldsymbol{x}))|(\boldsymbol{x}, y) \in \mathcal{D}'_s \wedge r(\boldsymbol{x}) \neq y\}$. Note that the proportion of stamped samples is typically lower than the detection rate $\mu = |\mathcal{D}'_s|/|\mathcal{D}_p|$, as we avoid stamping samples whose labels remain unchanged to let the backdoor take effect. Merging $\mathcal{D}'_m$ with data that are not stamped,

we obtain the processed dataset D'_p for training. The defense framework can be implemented during data preprocessing, without any modification to the end-to-end training pipeline. During inference, we stamp all inputs for defense.

In NAB, we further decompose the attack loss in Eq. (1) into the original attack loss and the *defense loss*:

$$\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_m}[\ell(f_\theta(\boldsymbol{x}),y)] + \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}'_m}[\ell(f_\theta(\boldsymbol{x}),y)] \quad (2)$$

Jointly optimizing the model using the three losses leaves two backdoors in the network: an adversarial backdoor triggered by $\mathcal{P}(\cdot)$ and a non-adversarial backdoor triggered by $\mathcal{S}(\mathcal{P}(\cdot))$. The non-adversarial one prevents a poisoned sample with stamp from being classified to the target class. Typically, D'_s is a mixture of poisoned and clean samples due to mistakes of detection methods. When the detection accuracy is low, the non-adversarial backdoor might influence the performance on clean data. Further analysis is presented in Sec. 4.4.

3.3. Backdoor Detection and Poisoning Strategy

While attackers can select samples to poison randomly and simply label them with the target class, defenders need more deliberation on data selection and relabeling strategy.

Backdoor Detection. To create a backdoor targeting poisoned data, we detect a set of suspicious samples $\mathcal{D}'_s$ from $\mathcal{D}_p$ with ratio μ. *Detection accuracy* is the ratio of poisoned samples in $\mathcal{D}'_s$. Different from detection-based defenses that aims at filtering all the poisoned samples, μ is typically smaller than the poisoning rate λ in NAB as we only need part of the poisoned data for backdoor injection.

Poisoning Strategy. The stamping function $\mathcal{S}(\cdot)$ is less important as long as it is perceptible to neural networks. We care more about the relabeling function $r(\cdot)$ which generates pseudo labels to approximate true labels. Although randomly generated pseudo labels suffices to create the non-adversarial backdoor, a higher *pseudo label accuracy* can help preserve the performance on clean data when the detection method malfunctions.

Many existing or naive methods can fulfill the relabeling and simplified backdoor detection tasks effectively [11, 23, 14]. Nevertheless, the NAB framework is independent of any specific detection method or poisoning strategy. As the chasing game between backdoor attack and defense goes on, stronger methods are likely to show up in the future, and NAB can be easily instantiated with the latest techniques. The flexibility and portability ensures the long-term value of our framework.

3.4. Test Data filtering

A wide range of previous works consider minimizing the attack success rate as their only goal on poisoned samples. However, misclassification might still happen even if the image is not classified to the target class, bringing about unintended consequences. If we add a requirement to the threat model that all poisoned test samples should be either identified or correctly classified, the defense effectiveness of some existing methods will be less satisfying.

An additional benefit of NAB is that it can be easily augmented with a test data filtering technique. Ideally, the prediction results on a clean sample $\boldsymbol{x}$ and its stamped version $\mathcal{S}(\boldsymbol{x})$ are both the true label y. However, the model tends to predict $c(y)$ on $\mathcal{P}(\boldsymbol{x})$ and $r(\mathcal{P}(\boldsymbol{x}))$ on $\mathcal{S}(\mathcal{P}(\boldsymbol{x}))$, which are expected to be different due to the defender's backdoor. Based on the observation above, we identify samples with $f_\theta(\boldsymbol{x}) \neq f_\theta(\mathcal{S}(\boldsymbol{x}))$ as poisoned and reject them during inference. In this way, the augmented NAB can handle poisoned data appropriately with a high accuracy.

4. Experiments

4.1. Experiment Settings

Attack. Experiments are conducted under 5 representative backdoor attacks, including two classical attacks: BadNets attack (patch-based) [10] and Blend attack (blending-based) [5], two advanced attacks: Dynamic attack (sample-specific) [30] and WaNet attack (invisible) [29], and one label-consistent attack: Clean-Label attack [40]. We follow the configurations suggested in the original papers, including the trigger patterns and trigger sizes. Performance of the attacks are evaluated on two datasets: CIFAR-10 (10 classes, 50k samples) [16] and tiny-ImageNet (200 classes, 100k samples) [17]. Dynamic attack and Clean-Label attack (CL) are omitted on tiny-ImageNet for failure of reproduction. The target label is set to 0 for both datasets. We set the poisoning rate $\lambda = 0.1$ for the first four attacks, and $\lambda = 0.25$ (2.5% of the whole training set) for CL.

Defense and Training. We instantiate the NAB framework with 3 backdoor detection techniques and 2 relabeling strategies throughout our experiments. The detection rate μ is set to 0.05 for the following methods:

- **Local Gradient Ascent (LGA)** [23]: Train with a tailored loss function in early epochs and isolate samples with lower training losses.
- **Label-Noise Learning (LN)** [14]: Train a classier appended to a self-supervised learning (SSL) pertained feature extractor with label-noise learning [42], and capture low-credible samples.
- **SPECTRE** [11]: Use robust covariance estimation to amplify the spectral signature of poisoned data and detect them with QUantum Entropy (QUE) scores.

In our poisoning strategy, $\mathcal{S}(\cdot)$ simply applies a 2×2 patch with value 0 on the upper left corner of the samples. We adopt the following strategies for pseudo label generation:

- **Verified Data (VD)**: Train a label predictor with supervised learning on a small collection of verified data (5% of the training set as assumed in [24]).
- **Nearest-Center (NC)**: Obtain representations using a SSL-pretrained model and assign pseudo labels according to the nearest center.

We *do not assume* that the methods above are state-of-the-art. They are chosen for their simplicity and can be safely replaced with comparable or stronger methods. LN and NC are introduced because one of our baselines [14] relies on a SSL stage. Experiments are conducted on ResNet-18 (by default) and ResNet-50 [12]. We train the models for 100 epochs with three data augmentations: random crop, horizontal flipping and cutout. The optimizer is Stochastic Gradient Descent (SGD) with momentum 0.9. Learning rate is set to 0.1 and decays with the cosine decay schedule [27]. More details are presented in the supplementary material.

Baselines. We compare our method with 3 state-of-the-art defense methods: 1) Neural Attention Distillation (NAD) [24] uses 5% of clean training data to fine-tune a student network under the guidance of a teacher model. We use the same set of verified data for NAD and the relabeling strategy VD. 2) Anti-Backdoor Learning (ABL) [23] unlearns the backdoor using a small set of isolated data. Note that an additional fine-tuning stage is added before backdoor unlearning to improve clean accuracy for fair comparison. 3) Decoupling-Based backdoor Defense (DBD) [14] divides the training pipeline into a three stages to prevent learning the backdoor. Despite its impressive performance under some attacks, we find that DBD fails when the poisoned samples are clustered after the self-supervised learning stage under some attacks (*e.g.* Dynamic attack [30]). Besides, DBD was tested without applying trigger patterns to the target class, but its performance drops when the constraint is removed. We leave a detailed discussion of the weaknesses in the supplementary material, and provide a separate comparison with our method following their original settings except for the poisoning rate.

Metrics. We adopt two widely used metrics for the main results: attack success rate (ASR, ratio of poisoned samples mistakenly classified to the target class) and Clean Accuracy (CA, ratio of correctly predicted clean samples). To test the effectiveness of our data filtering method, we further introduce backdoor accuracy (BA, ratio of correctly predicted backdoor samples), ratio of rejected clean data (C-REJ), prediction success rate (PSR, ratio of correctly predicted *and not* rejected clean samples), ratio of rejected poisoned data (B-REJ), and defense success rate (DSR, ratio of correctly predicted *or* rejected poisoned samples)

4.2. Main Results

Backdoor Detection and Pseudo Labels. LGA is adopted for backdoor detection in this section. As shown in Fig. 3, detection accuracy approaches its maximal value in most cases. However, the method has less satisfying performance under WaNet attack, which adopts a noise mode to escape detection. We generate pseudo labels with VD and NC. Fig. 3 shows that the latter method based on self-supervised learning generates pseudo labels of higher quality. In some cases, NC achieves very high accuracy on detected samples. We attribute this partially to the fact that LGA prefers to isolate images whose losses drop faster. These samples have more salient class features and are thus closer to the corresponding centers. On tiny-ImageNet, detection accuracy approaches 100%, but the label accuracy of VD is 46.64%, 38.90%, 42.34% for BadNets, Blend, WaNet respectively, posing a greater challenge than CIFAR-10.

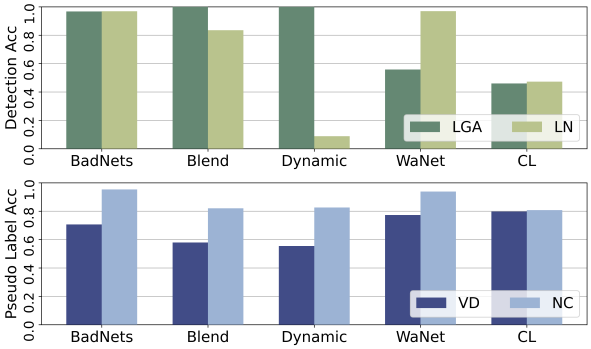


Figure 3: Detection and pseudo label accuracy on CIFAR-10. The maximal detection accuracy for CL attack is $\min(\frac{\lambda}{\mu}, 1) = 0.5$. Pseudo label accuracy is calculated on LGA detected samples.

Comparison with NAD and ABL. As shown in Tab. 1 and Tab. 2, NAB outperforms NAD and ABL by a large margin across all the settings in terms of clean accuracy. CA of our method is lower under WaNet and CL attack than other attacks because the a large number of clean samples are incorporated into the detection data $\mathcal{D}'_s$, but still higher than that of the baseline defenses. Our method also has outstanding performance in terms of attack success rate. It obtains the lowest ASR in most cases. On tiny-ImageNet, however, ABL achieves a significantly low ASR. We suspect that the diverse classes of the dataset help the unlearning stage of ABL identify backdoor features more precisely. We also find that results of our method are even better on ResNet-50, achieving a much lower ASR and a clean accuracy comparable to *no defense* in some cases. Model capacity might benefit the injection of an additional backdoor. In summary, our method suppresses the attacker's backdoor effectively while having limited influence on clean accuracy. By simply poisoning part of the training data, our method achieves state-of-the-art defense performance.

Comparison with DBD. DBD adopts self-supervised learning (SSL) in its first training stage, and part of the

Arch	ResNet-18								ResNet-50							
Defense	**No Defense**		**NAD**		**ABL**		**NAB (Ours)**		**No Defense**		**NAD**		**ABL**		**NAB (Ours)**	
Attack ↓	CA	ASR	CA	ASR	CA	ASR	CA	ASR	CA	ASR	CA	ASR	CA	ASR	CA	ASR
BadNets	93.99	100	89.09	2.04	91.85	**0.26**	**93.26**	0.93	94.09	99.52	89.97	1.29	92.80	0.50	**93.44**	**0.00**
Blend	94.09	100	89.29	1.22	89.87	1.62	**93.18**	**0.29**	94.26	99.98	90.04	1.03	88.11	1.41	**94.34**	**0.09**
Dynamic	94.29	99.99	89.11	10.28	91.64	1.74	**93.75**	**0.24**	94.00	99.98	89.80	4.53	92.50	1.30	**94.23**	**0.12**
WaNet	93.06	97.53	88.52	1.31	89.57	9.11	**90.36**	**0.67**	93.19	97.02	89.90	1.64	88.39	4.11	**91.54**	**0.34**
CL	94.66	99.73	88.97	4.63	87.27	0.61	**91.63**	**0.48**	94.70	91.58	90.16	4.22	88.24	0.98	**91.50**	**0.40**
Average	94.02	99.45	89.00	3.90	90.04	2.67	**92.44**	**0.52**	94.05	97.62	89.97	2.54	90.00	1.66	**93.01**	**0.19**

Table 1: Attack success rate (%) and clean accuracy (%) of NAD, ABL and our proposed method against 5 attacks over ResNet-18 and ResNet-50. The benchmark is CIFAR-10. We bold the best defense results under each attack.

Defense	No Defense		NAD		ABL		NAB (Ours)	
Attack ↓	CA	ASR	CA	ASR	CA	ASR	CA	ASR
BadNets	64.85	99.94	61.80	1.58	62.21	**0.00**	**63.14**	0.89
Blend	64.07	99.03	60.06	7.25	56.54	**0.00**	**63.04**	1.09
WaNet	64.33	99.86	60.16	3.91	55.79	0.69	**62.27**	**0.63**
Average	64.42	99.61	60.67	4.25	58.18	**0.23**	**62.81**	0.87

Table 2: Defense effectiveness (%) of baselines and our method against 3 attacks on tiny-ImageNet.

Defense	DBD		NAB (Ours)		NAB* (Ours)	
Attack ↓	CA	ASR	CA	ASR	CA	ASR
BadNets	92.60	1.49	93.69	**0.33**	**94.44**	0.42
Blend	92.64	1.87	94.18	0.48	**94.85**	**0.46**
WaNet	90.71	1.04	92.83	**0.54**	**93.55**	0.66
CL	92.94	0.95	93.83	1.31	**94.57**	**0.71**
Average	92.22	1.34	93.63	0.67	**94.35**	**0.56**

Table 3: Defense effectiveness (%) of self-supervised learning based defense methods on CIFAR-10.

impressive performance on clean accuracy comes from the extra training epochs. We also use the SSL pretrained network in pseudo label prediction (NC) and model initialization for fair comparison. As shown in Tab. 3, our method outperforms DBD by a large margin in both ASR and CA. NAB achieves better results even without SSL pre-training. We also find that higher pseudo label accuracy obtained by NC helps reduce the performance drop on clean data. More analyses of this factor are presented in Sec. 4.4.

4.3. Effectiveness of Data Filtering

We validate the effectiveness of data filtering and present the results in Tab. 4. For all the defenses listed, accuracy on

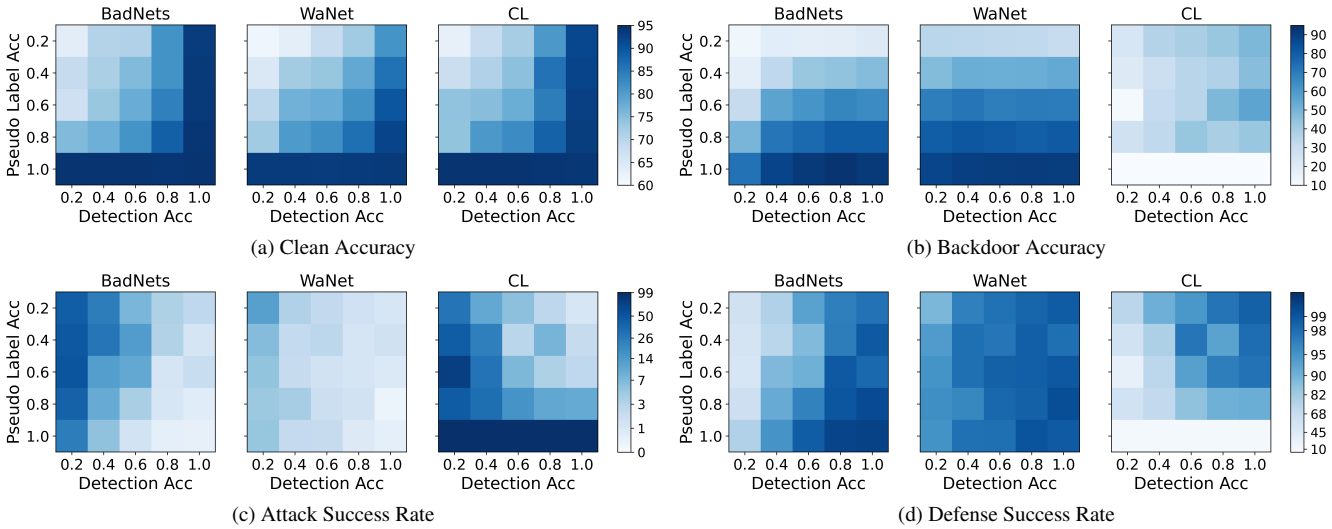
Defense	NAD	ABL	NAB	NAB + filtering			
Attack ↓	BA			C-REJ	PSR	B-REJ	DSR
BadNets	87.66	91.69	72.10	2.83	92.49	98.61	99.14
Blend	85.26	89.64	72.47	1.54	92.79	97.17	99.68
Dynamic	66.75	85.47	65.38	1.71	93.26	96.94	99.67
WaNet	86.51	81.48	79.51	6.47	89.04	89.30	99.15
CL	86.06	86.02	75.28	4.41	90.95	89.83	99.33
Average	82.45	86.86	72.95	3.39	91.71	94.37	99.39

Table 4: Backdoor accuracy (%) and effectiveness (%) of data filtering on CIFAR-10, ResNet-18.

poisoned data lags behind that on clean data. The gap is especially obvious for NAB since the model learns to predict a stamped sample to its pseudo label which is typically not very accurate. Augmenting NAB with data filtering provides a remedy for this. We find that the defense success rate reaches over 99% in all cases, suggesting that the filtering technique identifies most poisoned samples and those escaping from detection are typically correctly classified. Part of the clean samples are also rejected, but a significant performance drop is not observed. This is because most of the rejected clean samples are also misclassified.

4.4. Further Analyses

Detection and Pseudo Label Accuracy. Backdoor detection and pseudo label generation are two major components influencing the performance of our method. Analyses on them helps understand the robustness of NAB and can guide the selection of specific detection and relabeling strategies. Therefore, we test NAB under different detection accuracy (DA) and pseudo label accuracy (PLA), and present the results in Fig. 4. The following patterns can be observed: 1) Clean accuracy (CA) relies on both DA and PLA, and NAB can preserve a high CA when either DA or PLA reaches a



(a) Clean Accuracy (b) Backdoor Accuracy

(c) Attack Success Rate (d) Defense Success Rate

Figure 4: Clean accuracy, backdoor accuracy, attack success rate and defense success rate (%) under different detection accuracy and pseudo label accuracy. The experiments are conducted on CIFAR-10 under BadNets, WaNet and CL. To generate pseudo labels of accuracy p, we randomly change $1-p$ of the true labels to a different class. For a detection accuracy q, we randomly select qN poisoned samples and $(1-q)N$ clean samples, where N is size of the training set.

λ	**0.01**		**0.05**		**0.10**		**0.20**	
Metric	CA	ASR	CA	ASR	CA	ASR	CA	ASR
$\mu\downarrow$	**BadNets**							
0.00	94.55	100	94.39	100	93.96	100	94.00	100
0.01	94.45	1.00	93.97	8.76	94.18	61.96	93.91	78.72
0.05	94.45	0.77	93.96	0.61	93.29	1.19	93.22	1.29
0.10	91.23	0.79	92.81	0.67	93.01	0.13	93.01	0.22
$\mu\downarrow$	**Dynamic**							
0.00	94.71	99.64	94.38	99.99	94.41	99.99	94.22	99.99
0.01	94.34	1.06	94.19	28.99	94.05	53.71	93.70	82.17
0.05	93.61	0.82	94.19	0.39	93.95	0.46	93.82	0.64
0.10	90.66	0.87	91.98	0.42	93.16	0.24	93.69	0.31

Table 5: Defense effectiveness (%) of out method under different detection rate μ and poisoning rate λ on CIFAR-10. SPECTRE and NC are adopted for detection and pseudo label generation, respectively.

high level. 2) Attack success rate (ASR) is more sensitive to DA, as the metric directly influences how many poisoned samples are stamped for non-adversarial backdoor injection. 3) Backdoor accuracy (BA) mainly depends on PLA, but defense success rate (DSR) is more sensitive to DA. In practice, it is typically easy to find a detection method with accuracy over 90%, but pseudo label accuracy varies in different scenarios. CL, which is representative of label-consistent attacks, shows different reactions to PLA. The attack does not change the labels of poisoned samples, so actually we need incorrect pseudo labels to break the backdoor correlation. It is also worth mentioning that accuracy cannot fully reflect the quality of backdoor detection and pseudo label generation. For example, a detection method might have *detection bias*, which means it has a preference for detecting poisoned samples with some explicit patterns. A strong detection bias might hamper the defense performance of NAB even under a high DA. We leave a discussion on this problem in the supplementary material.

Detection Rate and Poisoning Rate. The defender needs to specify the detection rate μ without being aware of the poisoning rate λ. We experiment with different μ and λ and display the results in Tab. 5. When $\mu > \lambda$, the detection accuracy drops below $\frac{\lambda}{\mu}$. Performance on clean data is hampered to some extent, which is consistent with the conclusion made in the previous paragraph. When $\mu \leq \lambda$, NAB demonstrates satisfying performance on both CA and ASR. However, the defense effectiveness decays significantly when we have $\mu \ll \lambda$. The injected non-adversarial backdoor is not strong enough to suppress the attacker's backdoor in this case. Typically, attackers choose a small λ to escape inspection. Our choice $\mu = 0.05$ in the main experiments suffices to handle most situations.

Effectiveness under All-to-All Attack. So far we have tested NAB under all-to-one attack (A2O), where all the poisoned samples are relabeled to a single target class. In this section we introduce all-to-all attack (A2A) where samples with different original labels have different target labels. As shown in Tab. 6, A2A is less effective than A2O in

Attack		BadNets		Blend	
Defense ↓	**Arch ↓**	CA	ASR	CA	ASR
None	ResNet-18	94.29	93.37	93.75	90.12
	ResNet-50	94.53	93.97	94.20	90.87
NAB	ResNet-18	93.24	2.66	93.28	1.48
	ResNet-50	94.36	1.28	94.61	1.19

Table 6: Attack effectiveness (%) of all-to-all attacks and defense effectiveness (%) of our method under them on CIFAR-10. LN and NC are adopted for detection and pseudo label generation respectively.

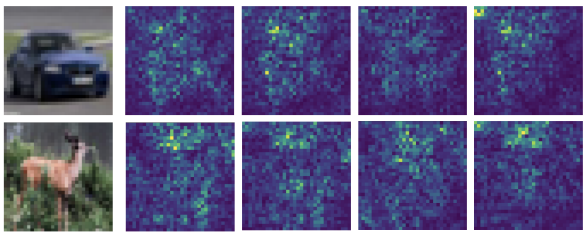


Figure 5: Examples of (a) raw images and saliency maps of their (b) clean, (c) stamped clean, (d) poisoned, (e) stamped and poisoned versions, which are obtained using NAB under BadNets attack.

terms of ASR. Our method can successfully defend against A2A, but the defense effectiveness is slightly lower than under A2O. Besides, it can be found that model capacity brings more benefits under A2A. We attribute it to that larger networks provide more learning ability to handle the complexified tasks in Eq. (1) and Eq. (2).

4.5. Understanding Non-Adversarial Backdoor

To get a comprehensive understanding of how NAB works, we first visualize the saliancy maps to illustrate how much attention the models pay on particular area of the input images. As shown in Fig. 5, the stamp (a 2×2 patch on the upper left corner) catches much attention when the trigger pattern is also added, but has a much weaker influence on clean data. This is consistent with the observation that the stamp can greatly change the behavior of a model only when the inputs are poisoned. We also visualize the representations in Fig. 1 to have a deeper insight into the mechanism behind our defense. Representations of stamped samples and clean samples are mixed up together, while those of poisoned samples are clearly separated except on the target class. These findings further demonstrate that our defense does not actually mitigate the attacker's backdoor, but inject a non-adversarial backdoor to suppress it. Besides, the model can directly predict the authentic labels of poisoned samples given a set of accurate pseudo labels.

5. Future Exploration with NAB Framework

We stress that the value of NAB is not limited to its simplicity, flexibility and impressive performance. The framework introduces the idea of backdoor for defense, which we believe is worthy of further exploration just like backdoor attack. Our implementation of NAB is not claimed to be optimal, and a lot more efforts can be done to strengthen it. We only list a few directions due to page limitation:

Protection for clean samples. The detection and relabeling accuracy might be both low in some cases. The non-adversarial backdoor would then be triggered on some clean samples and brings performance drop. The problem will possibly be alleviated by providing a protection mechanism (*e.g.* stamping some clean stamps without relabeling).

Sample-efficient backdoor. Injecting backdoor in a sample-efficient way has been a hot topic in backdoor attack [45, 47]. The defender will also want to inject a backdoor strong enough for defense with as few samples as possible. A benefit of sample-efficient backdoor for NAB is that when the number of required samples is small enough, the detected samples can go through human inspection and relabeling, guaranteeing a high DA and PLA.

Backdoor vaccination. A even more interesting question is whether the defender can carry out a backdoor attack, defend against it using NAB, and generalize the defense effectiveness to other attacks. We test the idea under a quite limited setting where the target class is known. The results displayed in the supplementary material show that ASR of Blend and WaNet attack is to some extent hampered by the defense targeting BadNets. If the generalization ability of defender's backdoor is further improved, NAB can dispose of backdoor detection and pseudo label generation since it only needs to focus on the attack transparent to defender.

6. Conclusion

In this work, we propose a novel defense framework NAB which injects a non-adversarial backdoor targeting poisoned data. Following the procedures in backdoor attack, we detect a small set of suspicious samples and process them with a poisoning strategy. During inference, we keep the non-adversarial backdoor triggered to suppress the effectiveness of attacker's backdoor. Extensive experiments demonstrate that NAB can achieve successful defense with minor performance drop on clean data. NAB has long-term value both as a powerful defense method and as a potential research area. As a method, its components are highly replaceable and can be updated and optimized in the future. As a research area, we hope that stronger variants would be derived from the simple and flexible framework, just as what has happened in backdoor attack.

References

[1] Yossi Adi, Carsten Baum, Moustapha Cisse, Benny Pinkas, and Joseph Keshet. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 1615–1631, 2018.

[2] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33:12449–12460, 2020.

[3] Mauro Barni, Kassem Kallas, and Benedetta Tondi. A new backdoor attack in cnns by training set corruption without label poisoning. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 101–105. IEEE, 2019.

[4] 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.

[5] 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.

[6] Edward Chou, Florian Tramer, and Giancarlo Pellegrino. Sentinet: Detecting localized universal attacks against deep learning systems. In *2020 IEEE Security and Privacy Workshops (SPW)*, pages 48–54. IEEE, 2020.

[7] Joseph Clements and Yingjie Lao. Backdoor attacks on neural network operations. In *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 1154–1158. IEEE, 2018.

[8] Min Du, Ruoxi Jia, and Dawn Song. Robust anomaly detection and backdoor attack detection via differential privacy. *arXiv preprint arXiv:1911.07116*, 2019.

[9] Jacob Dumford and Walter Scheirer. Backdooring convolutional neural networks via targeted weight perturbations. In *2020 IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–9. IEEE, 2020.

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

[11] Jonathan Hayase, Weihao Kong, Raghav Somani, and Sewoong Oh. Spectre: Defending against backdoor attacks using robust statistics. In *International Conference on Machine Learning*, pages 4129–4139. PMLR, 2021.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[13] 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.

[14] Kunzhe Huang, Yiming Li, Baoyuan Wu, Zhan Qin, and Kui Ren. Backdoor defense via decoupling the training process. *arXiv preprint arXiv:2202.03423*, 2022.

[15] Sara Kaviani and Insoo Sohn. Defense against neural trojan attacks: A survey. *Neurocomputing*, 423:651–667, 2021.

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

[17] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

[18] Shaofeng Li, Minhui Xue, Benjamin Zi Hao Zhao, Haojin Zhu, and Xinpeng Zhang. Invisible backdoor attacks on deep neural networks via steganography and regularization. *IEEE Transactions on Dependable and Secure Computing*, 18(5):2088–2105, 2020.

[19] Wenshuo Li, Jincheng Yu, Xuefei Ning, Pengjun Wang, Qi Wei, Yu Wang, and Huazhong Yang. Hu-fu: Hardware and software collaborative attack framework against neural networks. In *2018 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pages 482–487. IEEE, 2018.

[20] Yuanchun Li, Jiayi Hua, Haoyu Wang, Chunyang Chen, and Yunxin Liu. Deeppayload: Black-box backdoor attack on deep learning models through neural payload injection. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, pages 263–274. IEEE, 2021.

[21] Yiming Li, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[22] Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu. Invisible backdoor attack with sample-specific triggers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16463–16472, 2021.

[23] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Anti-backdoor learning: Training clean models on poisoned data. *Advances in Neural Information Processing Systems*, 34:14900–14912, 2021.

[24] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Neural attention distillation: Erasing backdoor triggers from deep neural networks. *arXiv preprint arXiv:2101.05930*, 2021.

[25] Yiming Li, Ziqi Zhang, Jiawang Bai, Baoyuan Wu, Yong Jiang, and Shu-Tao Xia. Open-sourced dataset protection via backdoor watermarking. *arXiv preprint arXiv:2010.05821*, 2020.

[26] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. 2017.

[27] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[28] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1765–1773, 2017.

[29] Anh Nguyen and Anh Tran. Wanet–imperceptible warping-based backdoor attack. *arXiv preprint arXiv:2102.10369*, 2021.

[30] Tuan Anh Nguyen and Anh Tran. Input-aware dynamic backdoor attack. *Advances in Neural Information Processing Systems*, 33:3454–3464, 2020.

[31] Han Qiu, Yi Zeng, Shangwei Guo, Tianwei Zhang, Meikang Qiu, and Bhavani Thuraisingham. Deepsweep: An evaluation framework for mitigating dnn backdoor attacks using data augmentation. In *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*, pages 363–377, 2021.

[32] Adnan Siraj Rakin, Zhezhi He, and Deliang Fan. Tbt: Targeted neural network attack with bit trojan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13198–13207, 2020.

[33] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

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

[35] Shawn Shan. Using honeypots to catch adversarial attacks on neural networks. In *Proceedings of the 8th ACM Workshop on Moving Target Defense*, pages 25–25, 2021.

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

[37] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014.

[38] Di Tang, XiaoFeng Wang, Haixu Tang, and Kehuan Zhang. Demon in the variant: Statistical analysis of {DNNs} for robust backdoor contamination detection. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 1541–1558, 2021.

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

[40] Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Clean-label backdoor attacks. 2018.

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[42] Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 322–330, 2019.

[43] Cheng-Hsin Weng, Yan-Ting Lee, and Shan-Hung Brandon Wu. On the trade-off between adversarial and backdoor robustness. *Advances in Neural Information Processing Systems*, 33:11973–11983, 2020.

[44] Dongxian Wu and Yisen Wang. Adversarial neuron pruning purifies backdoored deep models. *Advances in Neural Information Processing Systems*, 34:16913–16925, 2021.

[45] Pengfei Xia, Ziqiang Li, Wei Zhang, and Bin Li. Data-efficient backdoor attacks. *arXiv preprint arXiv:2204.12281*, 2022.

[46] Wayne Xiong, Jasha Droppo, Xuedong Huang, Frank Seide, Mike Seltzer, Andreas Stolcke, Dong Yu, and Geoffrey Zweig. Achieving human parity in conversational speech recognition. *arXiv preprint arXiv:1610.05256*, 2016.

[47] Yi Zeng, Minzhou Pan, Hoang Anh Just, Lingjuan Lyu, Meikang Qiu, and Ruoxi Jia. Narcissus: A practical clean-label backdoor attack with limited information. *arXiv preprint arXiv:2204.05255*, 2022.

[48] Pu Zhao, Pin-Yu Chen, Payel Das, Karthikeyan Natesan Ramamurthy, and Xue Lin. Bridging mode connectivity in loss landscapes and adversarial robustness. In *International Conference on Learning Representations*, 2020.

[49] Shihao Zhao, Xingjun Ma, Yisen Wang, James Bailey, Bo Li, and Yu-Gang Jiang. What do deep nets learn? class-wise patterns revealed in the input space. *arXiv preprint arXiv:2101.06898*, 2021.

[50] Shihao Zhao, Xingjun Ma, Xiang Zheng, James Bailey, Jingjing Chen, and Yu-Gang Jiang. Clean-label backdoor attacks on video recognition models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14443–14452, 2020.