Backdoor Defense via Adaptively Splitting Poisoned Dataset

Kuofeng Gao[1]*, Yang Bai [2]*, Jindong Gu[3]†, Yong Yang[4], Shu-Tao Xia[15]†
[1] Tsinghua University [2] Tencent Security Zhuque Lab [3] University of Oxford
[4] Tencent Security Platform Department [5] Peng Cheng Laboratory
gkf21@mails.tsinghua.edu.cn, {mavisbai,coolcyang}@tencent.com
jindong.gu@eng.ox.ac.uk, xiast@sz.tsinghua.edu.cn

Abstract

Backdoor defenses have been studied to alleviate the threat of deep neural networks (DNNs) being backdoor attacked and thus maliciously altered. Since DNNs usually adopt some external training data from an untrusted third party, a robust backdoor defense strategy during the training stage is of importance. We argue that the core of training-time defense is to select poisoned samples and to handle them properly. In this work, we summarize the training-time defenses from a unified framework as splitting the poisoned dataset into two data pools. Under our framework, we propose an adaptively splitting dataset-based defense (ASD). Concretely, we apply loss-guided split and meta-learning-inspired split to dynamically update two data pools. With the split clean data pool and polluted data pool, ASD successfully defends against backdoor attacks during training. Extensive experiments on multiple benchmark datasets and DNN models against six state-of-the-art backdoor attacks demonstrate the superiority of our ASD. Our code is available at https://github.com/KuofengGao/ASD.

1. Introduction

Backdoor attacks can induce malicious model behaviors by injecting a small portion of poisoned samples into the training dataset with specific trigger patterns. The attacks have posed a significant threat to deep neural networks (DNNs) [31–33, 35, 37, 45], especially when DNNs are deployed in safety-critical scenarios, such as autonomous driving [10]. To alleviate the threats, backdoor defenses have been intensively explored in the community, which can be roughly grouped into post-processing defenses and training-time ones. Since the training data collection is usually time-consuming and expensive, it is common to use external data for training without security guarantees [16, 24, 29, 42, 43]. The common practice makes backdoor attacks feasible in real-world applications, which highlights the importance of training-time defenses. We argue that such defenses are to solve two core problems, *i.e.*, to select poisoned samples and to handle them properly.

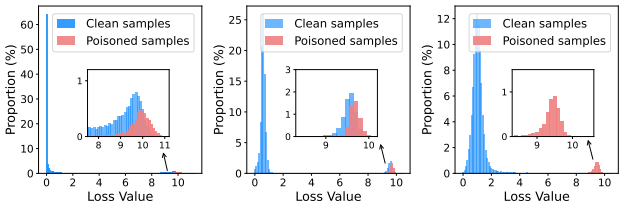
Table 1. Summary of the representative training-time backdoor defenses under our framework.

Methods	# Pool Initialization	# Pool Maintenance	# Pool Operation	# Clean Hard Sample Selection
ABL	Fast	Static	Unlearn	No
DBD	Slow	Adaptive	Purify	No
ASD (Ours)	Fast	Adaptive	Purify	Yes

In this work, we formulate the training-time defenses into a unified *framework* as splitting the poisoned dataset into two data pools. Concretely, a clean data pool contains selected clean samples with trustworthy labels and a polluted data pool is composed of poisoned samples and remaining clean samples. Under this framework, the mechanisms of these defenses can be summarized into three parts, *i.e.*, *pool initialization*, *pool maintenance*, and *pool operation*. To be more specific, they need to first initialize two data pools, deploy some data pool maintenance strategies, and take different training strategies on those split (clean and polluted) data pools. We illustrate our framework with two representative training-time defenses, *i.e.*, anti-backdoor learning (ABL) [27] as well as decoupled-based defense (DBD) [21]. ABL statically initializes a polluted pool by the loss-guided division. The polluted pool is fixed and unlearned during training. Similarly, DBD initializes two data pools after computationally expensive training. Then the model is fine-tuned by semi-supervised learning with two dynamically updated data pools. (More details of these two methods are introduced in Sec. 2.2.)

Despite their impressive results, there is still room for improvement. ABL initializes two pools with static data selection, which raises the concern of mixing poisoned data into clean ones. Once they are mixed, it is hard to alleviate it in the followed training process. Besides, unlearning poi-

*Equal contribution.
†Corresponding author.



(a) ABL (b) DBD (c) ASD (Ours)

Figure 1. Loss distribution of samples on the model trained by ABL, DBD and our ASD against WaNet. Compared with ABL and DBD, our proposed ASD can clearly separate clean samples and poisoned ones better by a novel meta-split.

soned data directly can lose some useful semantic features and degrade the model's clean accuracy. As for DBD, its pool initialization is computationally expensive and is hard to be implemented end-to-end. Moreover, DBD adopts supervised learning for the linear layer in the whole poisoned dataset during the second stage, which can potentially implant the backdoor in models.

Under our framework, we introduce an adaptively splitting dataset-based defense (ASD). With two initialized data pools, we first adopt the loss-guided [51] split to update two data pools. However, some (model-dependent) clean hard samples can not be distinguished from poisoned ones directly by their loss magnitudes. As shown in Fig. 1, ABL and DBD adopting loss-guided split have failed to completely separate clean samples from poisoned samples. Instead, we propose a novel meta-learning-inspired split (meta-split), which can make a successful separation. Then we treat the clean data pool as a labeled data container and the polluted one as unlabeled, where we adopt semi-supervised learning on two data pools. As such, we can utilize the semantic information of poisoned data without labels to keep the clean accuracy meanwhile to avoid backdoor injection, which can be regarded as a fashion of purifying poisoned samples. Note that, our ASD introduces clean seed samples (*i.e.*, only 10 images per class) in pool initialization, which could be further extended to a transfer-based version, by collecting clean seed samples from another classical dataset. Given previous methods [28, 34, 50, 54] usually assume they can obtain much more clean samples than ours, our requirements are easier to meet. The properties of ABL, DBD and our ASD are briefly listed in Table 1.

In summary, our main contributions are three-fold:

- We provide a framework to revisit existing training-time backdoor defenses from a unified perspective, namely, splitting the poisoned dataset into a clean pool and a polluted pool. Under our framework, we propose an end-to-end backdoor defense, ASD, via splitting poisoned dataset adaptively.
- We propose a fast pool initialization method and adaptively update two data pools in two splitting manners, *i.e.*, loss-guided split and meta-split. Especially, the proposed meta-split focuses on how to mine clean hard samples and clearly improves model performance.
- With two split data pools, we propose to train a model on the clean data pool with labels and the polluted data pool without using labels. Extensive experiment results demonstrate the superiority of our ASD to previous state-of-the-art backdoor defenses.

2. Related Work

2.1. Backdoor Attack

Backdoor attacks are often implemented by injecting a few poisoned samples to construct a poisoned dataset. Once a model is trained on the constructed poisoned dataset, the model will perform the hidden backdoor behavior, *e.g.*, classifying samples equipped with trigger to the target label. Apart from the malicious behavior, the backdoored model behaves normally when the trigger is absent. Existing backdoor attacks can be divided into two categories: (1) **poison-label backdoor attacks** [1, 16, 30, 53, 60] connect the trigger with the target class by relabelling poisoned samples as target labels. Trigger patterns have been designed to enhance the attack strength [16, 40] or stealthiness [8, 36, 39]. (2) **clean-label backdoor attacks** [14, 48] only poison the samples from the target class with labels unchanged. Although they are stealthier than poison-label attacks, clean-label attacks may fail to implant the model sometimes [25, 59].

2.2. Backdoor Defense

Recent work has explored various methods to mitigate the backdoor threat. Existing backdoor defenses can be grouped into two categories: (1) **post-processing backdoor defenses** [15, 23, 56] aim to repair a local backdoored model with a set of local prepared data. A straightforward way is to reconstruct the trigger pattern [6, 11, 17, 46, 49] and then to unlearn the trigger pattern for repairing the model. Apart from the trigger-synthesis defenses, other methods are also widely used in erasing the backdoor, *e.g.*, pruning [34, 54, 62], model distillation [28] and mode connectivity [58]. (2) **training-time backdoor defenses** [7, 47, 52] intend to train a clean model directly on the poisoned dataset. Anti-backdoor learning (ABL) [27] observes that the training losses of poisoned samples drop abruptly in the early training stage. Thus ABL proposes to first isolate a few samples with the lowest losses in early epochs, then train a model without isolated samples and finally unlearn isolated samples during the last few training epochs. Decouple-based backdoor defense (DBD) [21], first adopts self-supervised learning to obtain the feature extractor. Then it uses supervised learning to update the linear layer. After identifying clean samples by the symmetric cross-entropy loss, DBD conducts semi-supervised learning on the labeled clean data and unlabeled remaining data.

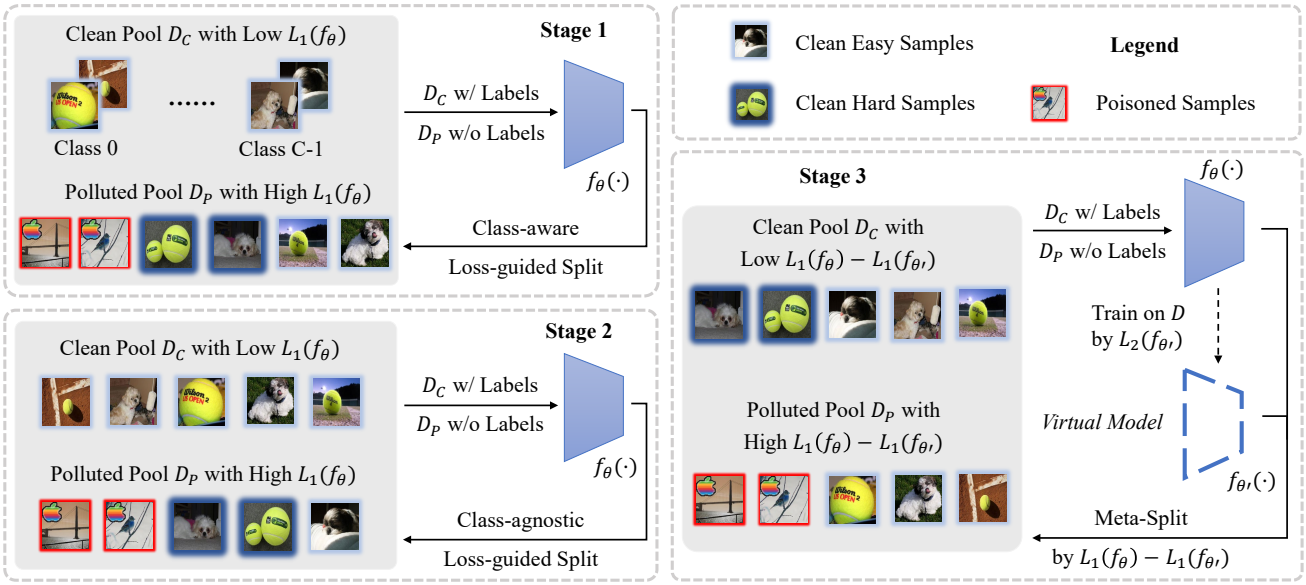


Figure 2. Pipeline of our ASD. It mainly contains three stages. (1) Clean data pool $\mathcal{D}_C$ is initialized with a few clean seed samples. Polluted data pool $\mathcal{D}_P$ is initialized with the poisoned training dataset $\mathcal{D}$. Then $\mathcal{D}_C$ is supplemented by samples in $\mathcal{D}$ with the lowest $\mathcal{L}_1(f_{\boldsymbol{\theta}})$ losses, which are equally selected from each class, dubbed *class-aware loss-guided split*. (2) $\mathcal{D}_C$ is substantially supplemented by samples with the lowest $\mathcal{L}_1(f_{\boldsymbol{\theta}})$ losses in the entire dataset $\mathcal{D}$, dubbed *class-agnostic loss-guided split*. (3) A novel *meta-split* is proposed to add clean hard samples into $\mathcal{D}_C$ and further improve the performance. Note that two data pools are adaptively updated at every epoch, where $\mathcal{D}_C$ is directly supplemented and the remaining samples in $\mathcal{D}$ are divided as $\mathcal{D}_P$. During the whole process, labels in $\mathcal{D}_C$ are preserved while labels in $\mathcal{D}_P$ are removed. The model $f_{\boldsymbol{\theta}}$ is trained on $\mathcal{D}_C$ and $\mathcal{D}_P$ by semi-supervised learning following Eq. 2.

Besides, adopting differential-privacy SGD [12] and strong data augmentation [4] can also defend against backdoor attacks to some degree. Our proposed ASD belongs to the training-time backdoor defense.

3. Preliminaries

Threat model. We adopt the poisoning-based threat model used in previous works [8, 16, 48], where attackers provide a poisoned training dataset containing a set of pre-created poisoned samples. Following previous training-time defenses [4, 12, 21, 27], we assume that defenders can control the training process. Besides, a few clean samples of each class are available as seed samples. The goal of defenders is to obtain a well-performed model without suffering backdoor attacks.

Problem formulation. Given a classification model $f_{\boldsymbol{\theta}}$ with randomly initialized parameters $\boldsymbol{\theta}$ and a training dataset $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$, the training dataset $\mathcal{D}$ contains N samples $\boldsymbol{x}_i \in \mathbb{R}^d, i = 1, ..., N$, and their ground-truth labels $y_i \in \{0, 1, ..., C-1\}$ where C is the number of classes. Poisoned samples might be included in $\mathcal{D}$. Under our unified framework, we propose to divide the dataset $\mathcal{D}$ into two disjoint data pools adaptively, *i.e.*, a clean data pool $\mathcal{D}_C$ with labels and a polluted data pool $\mathcal{D}_P$, whose labels will not be used. Moreover, we train the model on the clean data pool and polluted data pool in a semi-supervised learning-based manner by treating the polluted pool as unlabeled data, denoted as:

$$\min_{\boldsymbol{\theta}} \mathcal{L}\left(\mathcal{D}_C, \mathcal{D}_P; \boldsymbol{\theta}\right), \tag{1}$$

where $\mathcal{D}_C \subset \mathcal{D}$ and $\mathcal{D}_P = \{\boldsymbol{x} | (\boldsymbol{x}, y) \in \mathcal{D} \backslash \mathcal{D}_C\}$. $\mathcal{L}(\cdot)$ denotes the semi-supervised loss function. In view of previous semi-supervised learning methods [2,3,41,61], $\mathcal{L}(\cdot)$ usually contains two losses:

$$\mathcal{L} = \sum_{(\boldsymbol{x},y)\in\mathcal{D}_C} \mathcal{L}_s(\boldsymbol{x}, y; \boldsymbol{\theta}) + \lambda \sum_{\boldsymbol{x}\in\mathcal{D}_P} \mathcal{L}_u(\boldsymbol{x}; \boldsymbol{\theta}), \tag{2}$$

where the supervised loss $\mathcal{L}_s$ is adopted on the clean data pool $\mathcal{D}_C$ with labels, the unsupervised loss $\mathcal{L}_u$ is used on the polluted data pool $\mathcal{D}_P$ without using labels, λ denotes the trade-off between $\mathcal{L}_s$ and $\mathcal{L}_u$. Since $\mathcal{L}_s$ can obtain the precise relationship between images and labels. Hence, it is critical to ensure as many clean samples and few poisoned samples as possible in the clean data pool $\mathcal{D}_C$.

4. Proposed Backdoor Defense: ASD

In this section, we present the pipeline of our ASD. The key challenge is to adaptively update and maintain two pools, *i.e.*, a clean data pool $\mathcal{D}_C$ and a polluted data pool $\mathcal{D}_P$. As shown in Fig. 2, our ASD is summarized in three stages: (1) We first initialize $\mathcal{D}_C$ with several fixed clean seed samples and $\mathcal{D}_P$ with the whole poisoned dataset $\mathcal{D}$. We perform the warming-up training and update $\mathcal{D}_C$ by

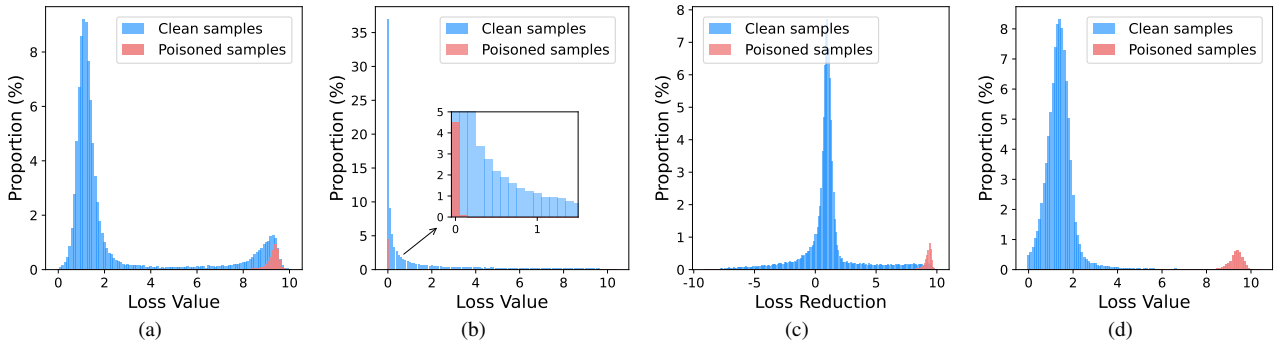


Figure 3. Loss distribution of samples on the model after different operations, which motivates us to propose the meta-split method in stage 3. (a) The model $f_{\boldsymbol{\theta}}$ after previous two stages. The clean hard samples have similar high losses to the poisoned ones, which can not be separated by the loss-guided split. (b) The 'virtual model' $f_{\boldsymbol{\theta}'}$ in Fig. 3a after one-epoch supervised learning. The losses of poisoned samples reduce to zero, which shows the poisoned samples can be fast learned. (c) The loss reduction between $f_{\boldsymbol{\theta}}$ in Fig. 3a and $f_{\boldsymbol{\theta}'}$ in Fig. 3b. (d) The model $f_{\boldsymbol{\theta}}$ after stage 3. Compared with Fig. 3a, poisoned samples and clean samples are better separated, especially for clean hard samples, which benefits from our proposed meta-split.

class-aware loss-guided split. (2) Then, we adopt class-agnostic loss-guided split on the entire dataset $\mathcal{D}$ and supplement $\mathcal{D}_C$ to accelerate the defense process. (3) To add more clean hard samples into $\mathcal{D}_C$, we propose a meta-learning-inspired (meta-split) method. During these three stages, $\mathcal{D}_P$ is composed of all remaining samples in $\mathcal{D}$ except for those selected in $\mathcal{D}_C$. The model $f_{\boldsymbol{\theta}}$ is trained by Eq. 2. Algorithm of our ASD is shown in Appendix A.

4.1. Warming-up and training with loss-guided split

We introduce the first two stages of ASD. Both two stages utilize the loss-guided data split to update the clean data pool $\mathcal{D}_C$ and the polluted data pool $\mathcal{D}_P$ adaptively. To be more specific, given a model $f_{\boldsymbol{\theta}}$ at any epoch during stage 1 and stage 2, $\mathcal{D}_C$ is supplemented by samples with the lowest $\mathcal{L}_1(f_{\boldsymbol{\theta}})$ losses, while $\mathcal{D}_P$ is composed of remaining samples. As suggested in [21], symmetric cross-entropy (SCE) [51] loss is adopted as $\mathcal{L}_1(\cdot)$ because it can amplify the difference between clean samples and poisoned samples when compared with cross-entropy (CE) loss.

Warming up with class-aware loss-guided split. In the warming-up stage, we first initial $\mathcal{D}_C$ with the clean seed samples and $\mathcal{D}_P$ with all the poisoned training data. Since only a few clean seed samples are available in $\mathcal{D}_C$, we progressively increase the number of samples in $\mathcal{D}_C$, namely we add n every t epochs in each class. Next, we add samples with the lowest $\mathcal{L}_1(\cdot)$ losses *in each class* to $\mathcal{D}_C$ dynamically, and remaining samples are used as $\mathcal{D}_P$. The reason for the class-aware way is to prevent the performance collapse caused by the class imbalance in the small clean data pool. Based on the tiny but progressively growing $\mathcal{D}_C$ and its complement $\mathcal{D}_P$, the model will be warmed up according to Eq. 2 during the first T_1 epochs.

Training with class-agnostic loss-guided split. After the first stage, the model with certain accuracy can be used to better distinguish clean samples and poisoned samples. In the second stage, we further enlarge $|\mathcal{D}_C|$ to accelerate the defense process. We directly add $\alpha\%$ samples with the lowest $\mathcal{L}_1(\cdot)$ losses in the entire dataset into $\mathcal{D}_C$, and remaining samples are used as $\mathcal{D}_P$. Such a class-agnostic loss-guided data split method can avoid selecting poisoned samples into $\mathcal{D}_C$ from the target class and further suppress the attack success rate. With two split data pools, we adopt Eq. 2 to train the model from epoch T_1 to epoch T_2.

4.2. Hard sample training with meta-split

After previous two stages, there is still a small gap in clean accuracy between our ASD defended model and the normally trained model (in Fig. 4). We assume that the gap is caused by that some clean samples can not be accessed by the loss-guided split. To verify our assumption, we calculate the losses of all training data for BadNets attack [16]. In Fig. 3a, though the losses of most clean samples are nearly zeros, there are still some clean samples with high losses, which are similar to poisoned ones. Such phenomena indicate that most clean data have been well learned while poisoned data are not by our model after previous two stages. However, there are still some *(model-dependent) clean hard samples*, which are difficult to be separated from poisoned ones just by loss magnitudes. We argue such clean hard samples are quite difficult to learn by a model yet poisoned samples can be easy to learn during supervised learning. Fig. 3b and Fig. 3c can verify our claim and show that losses of poisoned sam-

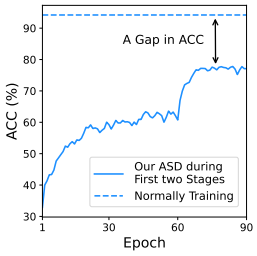


Figure 4. The clean accuracy of our ASD defended model during previous two stages and normally trained model.

ples drop much faster than those of clean hard samples after one-epoch supervised learning, which inspires us to provide a new solution to successfully separate clean hard samples and poisoned ones.

To distinguish clean hard samples from poisoned ones and inspired by meta-learning [13, 19], we propose a novel manner to supplement $\mathcal{D}_C$, called meta-split. Given a model $f_{\boldsymbol{\theta}}$ at any epoch in the third stage, we first create a new '*virtual model*' $f_{\boldsymbol{\theta}'}$ with the same parameters and architecture as $f_{\boldsymbol{\theta}}$. The virtual model $f_{\boldsymbol{\theta}'}$ is updated on the entire poisoned dataset $\mathcal{D}$ by the loss $\mathcal{L}_2(\cdot)$ with learning rate β which can be denoted as:

$$\begin{aligned} \boldsymbol{\theta}' &\leftarrow \boldsymbol{\theta}, \\ \boldsymbol{\theta}' &\leftarrow \boldsymbol{\theta}' - \beta \nabla_{\boldsymbol{\theta}'} \mathcal{L}_2(f_{\boldsymbol{\theta}'}(\boldsymbol{x}), y), \end{aligned} \tag{3}$$

where cross-entropy (CE) loss is adopted as $\mathcal{L}_2(\cdot)$. Finally, $\gamma\%$ samples with the least loss reduction $\mathcal{L}_1(f_{\boldsymbol{\theta}}) - \mathcal{L}_1(f_{\boldsymbol{\theta}'})$ are chosen to supplement $\mathcal{D}_C$.

Note that the virtual model $f_{\boldsymbol{\theta}'}$ is only used for sample separation and is not involved in the followed training process. After supplementing $\mathcal{D}_C$ by meta-split and consistently updating the model $f_{\boldsymbol{\theta}}$ using Eq. 2 until training end at epoch T_3, $f_{\boldsymbol{\theta}}$ can successfully split more clean hard samples in $\mathcal{D}_C$ and thus obtain higher accuracy. The loss distribution of samples on $f_{\boldsymbol{\theta}}$ at T_3 in Fig. 3d shows clean samples and poisoned samples have been successfully separated, which further demonstrates the effectiveness of meta-split during the third stage of our ASD.

5. Experiments

5.1. Experimental Setups

Datasets and DNN models. We adopt three benchmark datasets to evaluate all the backdoor defenses, *i.e.*, CIFAR-10 [9], GTSRB [44] and an ImageNet [9] subset. ResNet-18 [18] is set as the default model in our experiments. More details are listed in Appendix B.1. Moreover, we also provide the results on VGGFace2 dataset [5] and DenseNet-121 [20] in Appendix C.

Attack baselines and setups. We conduct six state-of-the-art backdoor attacks, including BadNets [16], Blend backdoor attack (Blend) [8], Warping-based backdoor attack (WaNet) [39], Input-aware backdoor attack (IAB) [40], Reflection-based attack (Refool) [36] and clean-label attack with adversarial perturbations (dubbed 'CLB') [48]. All these backdoor attacks are implemented as suggested in [21] and their original papers. Following [21], we choose 3 ($y_t = 3$) as the target label and set the poisoned rate as 25% poisoned samples in the target class for the clean-label attack and 5% for other five backdoor attacks. More details about the attack setups are summarized in Appendix B.2. Besides, we also provide the results against the sample-specific attack [26] and *all2all* attack in Appendix D.

Defense baselines and setups. We compare our proposed method with four existing backdoor defenses, including Fine-pruning (FP) [34], Neural Attention Distillation (NAD) [28], Anti-Backdoor Learning (ABL) [27] and decoupling-based backdoor defense (DBD) [21]. Since FP, NAD and ABL are sensitive to their hyper-parameters, we report their best results optimized by grid-search (See Appendix N). DBD is implemented based on the original paper [21]. Besides, FP and NAD are assumed to have access to 5% of the clean training data. Furthermore, we also provide the results of cutmix-based defense [4] and differential privacy SGD-based defense [12] in Appendix E.

For our ASD, we follow DBD [21] to adopt MixMatch [3] as our default semi-supervised method. The initial number of clean seed samples w is 10 in each class and it will increase $n = 10$ at every $t = 5$ epochs. After stage 1, the filtering rate $\alpha\%$ is set to 50%. For meta-split, we choose Adam [22] optimizer with the learning rate $\beta = 0.015$ to perform one-epoch supervised learning on a virtual model. In particular, we only update the parameters of the last three layers. The discussions about the hyper-parameters for meta-split are demonstrated in Appendix I. On CIFAR-10 and ImageNet, T_1, T_2 and T_3 are chosen as 60, 90 and 120, and T_3 is chosen as 100 on GTSRB. More details about the defense setups are in Appendix B.3.

Evaluation metrics. We evaluate backdoor defenses by two widely used metrics, *i.e.*, the accuracy on clean dataset (ACC) and the attack success rate (ASR). Specifically, ASR is the fraction of the samples from the non-target class with the trigger classified to the target label by the backdoored model. For a backdoor defense, the higher ACC and the lower ASR correspond to better performance.

5.2. Main Results

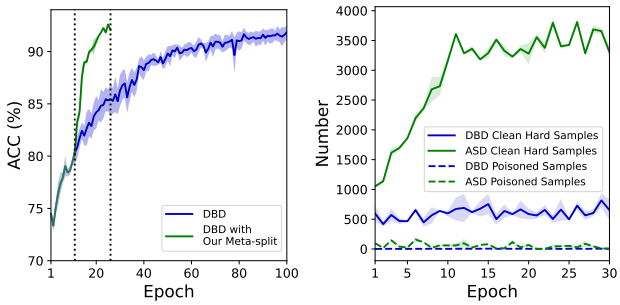
To verify the superiority of our backdoor defense, we summarize the ACC and ASR of five backdoor defenses against six backdoor attacks on three datasets in Table 2. We also report the time cost of three training-time defenses in Table 3. Table 2 shows that our ASD can achieve low ASRs meanwhile maintain high ACCs on three datasets. In particular, ASD can purify and better utilize poisoned samples, even outperforming 'No Defense' on ImageNet on average. In comparison, although FP and NAD require a larger amount of local clean samples (2,500) than ours (100), FP provides a limited reduction on ASR and NAD damages the ACC of the models by a large margin, which constrains their deployment in repairing the model. Table 3 shows that ABL requires the least training time among three training-time defenses but needs grid-search for hyper-parameters to defend against different attacks. Besides, when ABL encounters the WaNet or Refool, its static backdoor isolation samples can be mixed with some clean samples. Once ABL adopts these backdoor isolation samples to unlearn for

Table 2. The clean accuracy (ACC %) and the attack success rate (ASR %) of five backdoor defenses against six backdoor attacks across three datasets, including CIFAR-10, GTSRB and ImageNet. Note that the results of FP, NAD and ABL are reported by grid-searching the best ones in different hyper-parameters. In contrast, the results of DBD and our ASD are provided in the same hyper-parameters. Best results among five defenses are highlighted in **bold**.

Dataset	Attack	No Defense		FP		NAD		ABL		DBD		ASD (Ours)	
		ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR
CIFAR-10	BadNets	94.9	100	**93.9**	1.8	88.2	4.6	93.8	1.1	92.3	**0.8**	93.4	1.2
	Blend	94.1	98.3	92.9	77.1	85.8	3.4	91.9	1.6	91.7	**0.7**	**93.7**	1.6
	WaNet	93.6	99.9	90.4	98.6	71.3	6.7	84.1	2.2	91.4	**0**	**93.1**	1.7
	IAB	94.2	100	89.3	98.1	82.8	4.2	**93.4**	5.1	91.6	100	93.2	**1.3**
	Refool	93.8	98.2	92.1	86.1	86.2	3.6	82.7	1.3	91.5	0.5	**93.5**	**0**
	CLB	94.4	99.9	90.2	92.8	86.4	9.5	86.6	1.3	90.6	**0.1**	**93.1**	0.9
	Average	94.2	99.4	91.5	75.8	83.5	5.3	88.7	2.1	91.5	17.0	**93.3**	**1.1**
GTSRB	BadNets	97.6	100	84.2	**0**	**97.1**	0.2	**97.1**	**0**	91.4	**0**	96.7	**0**
	Blend	97.2	99.4	91.4	68.1	93.3	62.4	**97.1**	0.5	91.5	99.9	**97.1**	**0.3**
	WaNet	97.2	100	92.5	21.4	96.5	47.1	97.0	0.4	89.6	**0**	**97.2**	0.3
	IAB	97.3	100	86.9	**0**	97.1	0.1	**97.4**	0.6	90.9	100	96.9	**0**
	Refool	97.5	99.8	91.5	0.2	95.5	1.4	96.2	**0**	91.4	0.4	**96.8**	**0**
	CLB	97.3	100	93.6	99.3	3.3	21.1	90.4	2.3	89.7	0.3	**97.3**	**0**
	Average	97.4	99.9	90.0	31.5	80.5	22.1	95.9	0.6	90.8	33.4	**97.0**	**0.1**
ImageNet	BadNets	79.5	99.8	70.3	1.6	65.1	5.1	83.1	**0**	81.9	0.3	**83.3**	0.1
	Blend	82.5	99.5	63.4	9.5	64.8	0.3	**82.6**	0.7	82.3	100	82.5	**0.2**
	WaNet	79.1	98.9	58.2	84.4	63.8	1.3	74.9	1.1	80.6	9.8	**84.1**	**0.8**
	IAB	78.2	99.6	58.7	84.2	63.8	0.6	81.7	**0**	**83.1**	**0**	81.6	0.5
	Refool	80.6	99.9	61.4	10.3	63.7	0.3	76.2	0.2	82.5	0.1	**82.6**	**0**
	CLB	80.1	42.8	73.2	38.3	62.7	1.7	**82.8**	0.8	81.8	**0**	82.2	**0**
	Average	80.0	90.1	64.2	38.1	64.0	1.5	80.2	0.5	82.0	18.4	**82.7**	**0.3**

Table 3. The average training time (s) of ABL, DBD and our ASD.

Dataset	Method	Stage 1	Stage 2	Stage 3	Total
CIFAR-10	ABL	740	2,450	10	3,200
	DBD	30,000	80	15,770	45,850
	ASD (Ours)	4,980	2,490	2,520	9,990
GTSRB	ABL	520	1,750	5	2,275
	DBD	23,000	57	15,580	38,637
	ASD (Ours)	4,920	2,460	830	8,210
ImageNet	ABL	900	2,940	15	3,855
	DBD	180,000	350	42,750	223,100
	ASD (Ours)	13,500	6,750	6,780	27,030



(a) (b)

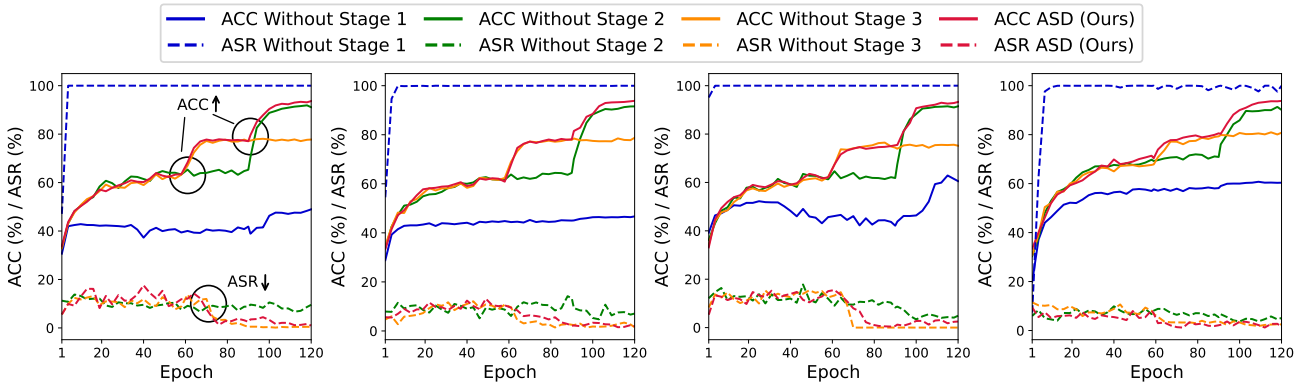
Figure 5. Combination and comparison between DBD and our ASD. The experiments (±std over 5 random runs) are conducted on CIFAR-10 for BadNets. (a) Combine DBD and our meta-split to converge faster. (b) The number of clean hard samples and poisoned samples in the clean data pool $\mathcal{D}_C$ during the final 30 epochs. Our ASD can successfully select clean hard examples.

the backdoored model, the ACC will drop. In contrast, our adaptive split can update two data pools dynamically, which stabilizes the defense process of our ASD.

DBD can achieve a low ASR in most cases but fails sometimes, *e.g.*, when it defends the IAB attack on CIFAR-10. The reason behind the observation is that DBD trains the linear layer on the whole poisoned dataset during stage 2 for 10 epochs, which introduces the risk to implant the backdoor. Besides, DBD needs a larger amount of time than our ASD, *e.g.*, 8 times our ASD on ImageNet. Stage 3 of DBD under our framework also splits the poisoned dataset into two data pools adaptively. Hence, after stage 3 of DBD for 10 epochs, we replace the original data split method in DBD with our proposed meta-split. As shown in Fig. 5a, stage 3 of DBD can be compressed from the original 190 epochs to 25 epochs and with 91+% ACC and 4−% ASR, which shows that our meta-split can help accelerate DBD. Moreover, we compare the number of clean hard samples and poisoned samples in $\mathcal{D}_C$ during the final 30 epochs of DBD and our ASD. Specifically, we choose 5,000 samples with the largest $\mathcal{L}_1(\cdot)$ losses chosen by the model as the clean hard samples. Fig. 5b demonstrates that our ASD can access much more clean hard samples than DBD and poisoned samples with a similar low scale. More results about the combination and comparison between DBD and our ASD are in Appendix F and G.

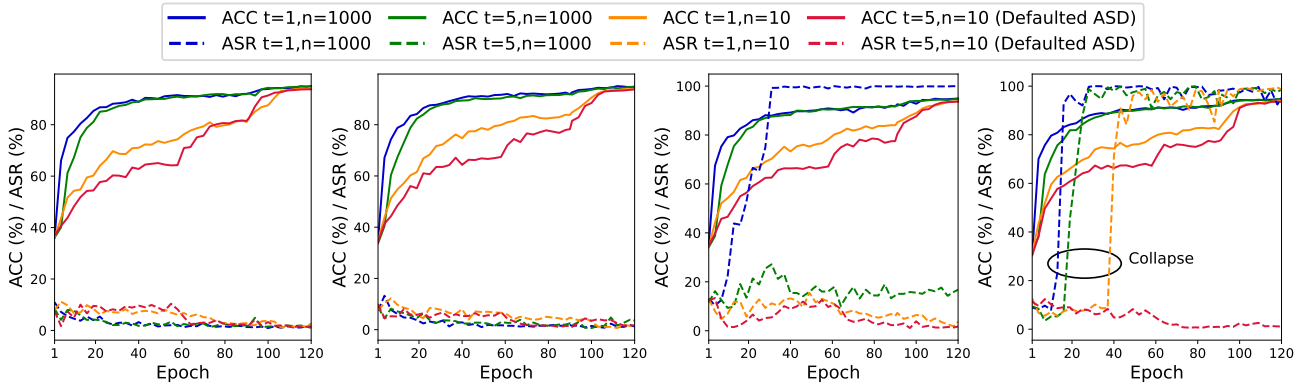
5.3. Ablation Study on Defense Settings

In summary, ASD is composed of three stages. Here we study the necessity of each stage by conducting the following experiments. Results are shown in Fig. 6. We set the poisoned rate of BadNets, Blend and WaNet as 20% and remain other settings unchanged. (1) **Without Stage 1**. The



(a) BadNets (b) Blend (c) WaNet (d) CLB

Figure 6. The clean accuracy (ACC %) and the attack success rate (ASR %) of different ablation studies for three stages in ASD on CIFAR-10 for four backdoor attacks, which shows the necessity of each stage in our ASD.



(a) BadNets (b) Blend (c) WaNet (d) CLB

Figure 7. The clean accuracy (ACC %) and the attack success rate (ASR %) of different warming-up strategies in stage 1 on CIFAR-10 for four backdoor attacks. A smaller t and a larger n correspond to a faster warming-up. The t and n should be set carefully to progressively increase the number of samples in $\mathcal{D}_C$ during stage 1 instead of building it in short time, which can prevent the collapse of ASD.

two data pools will approximate random initialization if we directly start our defense without stage 1. Fig. 6 indicates that the defense process will be completely disrupted with two randomly initialized data pools. (2) **Without Stage 2**. In stage 2, since less poisoned samples from the target class will be introduced to a larger $\mathcal{D}_C$ by class-agnostic loss-guided split, it can rapidly promote the ACC and suppress the ASR by a large margin. The defaulted ASD can achieve a lower ASR and a higher ACC than that without stage 2. (3) **Without stage 3**. As shown in Fig. 6, ACC will achieve only about 80% without stage 3 owing to the lack of the model-dependent clean hard samples in $\mathcal{D}_C$. More results about the ablation study on attack settings and defense settings are shown in Appendix H and I.

Different warming-up strategies. Compared with our default setting ($t = 5$ and $n = 10$), ACC can increase faster when building $\mathcal{D}_C$ in shorter time, *i.e.*, t is smaller and n is larger, as shown in Fig. 7. However, it can wrongly introduce a large number of poisoned samples into $\mathcal{D}_C$ and result in the failure of our ASD, especially under WaNet and CLB. Hence, it is necessary to control the speed to build $\mathcal{D}_C$ and constrain the number of samples in $\mathcal{D}_C$ during stage 1.

Different semi-supervised learning methods. We treat the samples in $\mathcal{D}_P$ as unlabeled and apply semi-supervised learning to learn from both data pools. In this experiment, we show our ASD can work well with various semi-supervised learning, *e.g.*, UDA [55] and ReMixMatch [2]. We keep all settings unchanged. As shown in Table 4, UDA and ReMixMatch can still have similar robustness against backdoor attacks compared with MixMatch [3] under our proposed ASD. More details about these three semi-supervised learning methods are in Appendix J.

5.4. Ablation Study on Seed Sample Selection

In our method, we utilize the seed samples with 10 clean samples per class to warm up the model. Here, we discuss the flexibility of the seed sample selection. (1) Seed samples contain a few poisoned samples. (2) Seed samples are from another classical dataset, *e.g.*, ImageNet.

First, we discuss the case that some poisoned samples are introduced in the seed samples and the results are shown in Table 5. It can be seen that our ASD can still exceed 91%

Table 4. The clean accuracy (ACC %) and the attack success rate (ASR %) on CIFAR-10 of our ASD implemented by different semi-supervised methods. Consistent satisfactory results show the stability of ASD.

Method	BadNets		Blend		WaNet		CLB	
	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR
MixMatch	93.4	1.2	93.7	1.6	93.1	1.7	93.1	0.9
UDA	92.6	2.1	91.9	2.3	92.5	1.6	92.1	2.8
ReMixMatch	91.6	1.4	91.5	1.0	91.9	0	91.1	0.1

Table 5. The clean accuracy (ACC %) and attack success rate (ASR %) on CIFAR-10 for different numbers of poisoned samples in the seed sample.

Poisoned Number		0	1	2	3	4
BadNets	ACC	93.4	94.1	93.6	93.6	93.5
	ASR	1.2	1.5	2.5	1.4	1.3
Blend	ACC	93.7	93.6	93.5	93.5	*93.1*
	ASR	1.6	2.5	2.7	0.8	*99.9*
WaNet	ACC	93.1	93.6	93.7	93.4	93.5
	ASR	1.7	1.4	2.2	4.1	5.4
CLB	ACC	93.1	93.5	91.3	93.7	*93.2*
	ASR	0.9	1.3	2.5	1.5	*98.9*

ACC and suppress the creation of backdoor even though the seed samples contain $1 \sim 3$ poisoned samples. Meanwhile, as the poisoned number increases to 4, our ASD can also defend against BadNets and WaNet successfully. This illustrates that our method has certain resistance to the seed samples mixed with a few poisoned samples. Then, we introduce the transfer learning-based ASD when adopting the seed samples from another classical dataset as follows.

Threat model. Considering a more realistic scenario, we cannot obtain any clean sample from the source dataset. Here, we specify the source training data as CIFAR-10. However, only 100 clean samples from the classical ImageNet dataset are available.

Methods. We first assign the 100 ImageNet clean samples as $\mathcal{D}_C$ and remove the labels of the entire poisoned CIFAR-10 as $\mathcal{D}_P$ and perform the semi-supervised learning for 10 epochs. Then we freeze the pre-trained backbone and fine-tune the linear layer on the entire poisoned dataset via supervised learning for 1 epoch. Finally, this model will be regarded as the initialized model of our ASD and other settings of our ASD remain unchanged.

Results. As shown in Table 6, after the above transfer-based pre-training, the model will achieve about 52% ACC and 9% ASR. By adopting this transfer-based initialized model, our ASD achieves $92+\%$ ACC and $4-\%$ ASR, which shows our ASD can obtain robustness against backdoor attacks without clean seed samples from the poisoned training dataset. More results are in Appendix I.

5.5. Resistance to Potential Adaptive Attacks

In the above experiments, we assume that attackers have no information about our backdoor defense. In this section, we consider a more challenging setting, where the attackers know the existence of our defense and can construct the poisoned dataset with an adaptive attack.

Table 6. The clean accuracy (ACC %) and the attack success rate (ASR %) on CIFAR-10 under the proposed transfer-based setting. Transfer learning-based ASD works well.

Method	BadNets		Blend		WaNet		CLB	
	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR
Transfer-based pre-training	52.6	8.9	51.5	10.2	53.4	9.3	52.6	9.5
Transfer-based ASD	92.9	2.4	92.5	2.6	92.5	3.5	92.1	2.8

Threat model for the attackers. Following existing work [8, 16, 48], we assume that the attackers can access the entire dataset and know the architecture of the victim model. However, the attackers can not control the training process after poisoned samples are injected into the training dataset.

Methods. Our defense separates samples by the magnitude of the loss reduction in the final stage, so adaptive attacks should aim to minimize the difference in the loss reduction between clean samples and poisoned samples. First, the attackers train a clean model in advance. Then, since the gradient determines the loss reduction of the model [22,57], the trigger pattern can be optimized by minimizing the gradient for poisoned samples *w.r.t* the trained model and maximizing that for clean samples.

Settings. We conduct experiments on CIFAR-10. Based on the clean model, we adopt projected gradient descent (PGD) [38] to optimize the trigger pattern for 200 iterations with a step size 0.001. Besides, we set the perturbation magnitude as $32/255$ and the trigger size as 32×32.

Results. The adaptive attack can achieve 94.8% ACC and 99.8% ASR without any defense. However, this attack can obtain 93.6% ACC and only 1.4% ASR under our ASD, which illustrates our defense can resist the adaptive attack. The probable reason is that the trigger pattern is optimized on the surrogate clean model and has low transferability. The details of this adaptive attack and another designed adaptive attack are stated in Appendix K and Appendix L.

6. Conclusion

In this paper, we revisit training-time backdoor defenses in a unified framework from the perspective of splitting the poisoned dataset into two data pools. Under our framework, we propose a backdoor defense via adaptively splitting the poisoned dataset. Extensive experiments show that our ASD can behave effectively and efficiently against six state-of-the-art backdoor attacks. Furthermore, we explore a transfer-based ASD to show the flexibility of seed sample selection in our method. In summary, we believe that our ASD can serve as an effective tool in the community to improve the robustness of DNNs against backdoor attacks.

References

[1] Jiawang Bai, Kuofeng Gao, Dihong Gong, Shu-Tao Xia, Zhifeng Li, and Wei Liu. Hardly perceptible trojan attack against neural networks with bit flips. In *ECCV*, 2022. 2

[2] David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. In *ICLR*, 2020. 3, 7

[3] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *NeurIPS*, 2019. 3, 5, 7

[4] Eitan Borgnia, Valeriia Cherepanova, Liam Fowl, Amin Ghiasi, Jonas Geiping, Micah Goldblum, Tom Goldstein, and Arjun Gupta. Strong data augmentation sanitizes poisoning and backdoor attacks without an accuracy tradeoff. In *ICASSP*, 2021. 3, 5

[5] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *FG*. IEEE, 2018. 5

[6] Tianlong Chen, Zhenyu Zhang, Yihua Zhang, Shiyu Chang, Sijia Liu, and Zhangyang Wang. Quarantine: Sparsity can uncover the trojan attack trigger for free. In *CVPR*, 2022. 2

[7] Weixin Chen, Baoyuan Wu, and Haoqian Wang. Effective backdoor defense by exploiting sensitivity of poisoned samples. In *NeurIPS*, 2022. 2

[8] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. 2017. 2, 3, 5, 8

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 5

[10] 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*, pages 299–318. Springer, 2019. 1

[11] Yinpeng Dong, Xiao Yang, Zhijie Deng, Tianyu Pang, Zihao Xiao, Hang Su, and Jun Zhu. Black-box detection of backdoor attacks with limited information and data. In *ICCV*, 2021. 2

[12] Min Du, Ruoxi Jia, and Dawn Song. Robust anomaly detection and backdoor attack detection via differential privacy. In *ICLR*, 2020. 3, 5

[13] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017. 5

[14] Leilei Gan, Jiwei Li, Tianwei Zhang, Xiaoya Li, Yuxian Meng, Fei Wu, Shangwei Guo, and Chun Fan. Triggerless backdoor attack for nlp tasks with clean labels. In *NAACL*, 2022. 2

[15] Yansong Gao, Change Xu, Derui Wang, Shiping Chen, Damith C Ranasinghe, and Surya Nepal. Strip: A defence against trojan attacks on deep neural networks. In *ACSAC*, 2019. 2

[16] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. In *IEEE Access*, 2019. 1, 2, 3, 4, 5, 8

[17] Jiyang Guan, Zhuozhuo Tu, Ran He, and Dacheng Tao. Few-shot backdoor defense using shapley estimation. In *CVPR*, 2022. 2

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5

[19] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(9):5149–5169, 2021. 5

[20] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, 2017. 5

[21] Kunzhe Huang, Yiming Li, Baoyuan Wu, Zhan Qin, and Kui Ren. Backdoor defense via decoupling the training process. In *ICLR*, 2022. 1, 2, 3, 4, 5

[22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 5, 8

[23] Soheil Kolouri, Aniruddha Saha, Hamed Pirsiavash, and Heiko Hoffmann. Universal litmus patterns: Revealing backdoor attacks in cnns. In *CVPR*, 2020. 2

[24] Yiming Li, Yang Bai, Yong Jiang, Yong Yang, Shu-Tao Xia, and Bo Li. Untargeted backdoor watermark: Towards harmless and stealthy dataset copyright protection. In *NeurIPS*, 2022. 1

[25] Yiming Li, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022. 2

[26] Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu. Invisible backdoor attack with sample-specific triggers. In *ICCV*, 2021. 5

[27] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Anti-backdoor learning: Training clean models on poisoned data. In *NeurIPS*, 2021. 1, 2, 3, 5

[28] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Neural attention distillation: Erasing backdoor triggers from deep neural networks. In *ICLR*, 2021. 2, 5

[29] Yiming Li, Mengxi Ya, Yang Bai, Yong Jiang, and Shu-Tao Xia. Backdoorbox: A python toolbox for backdoor learning. In *ICLR Workshop*, 2023. 1

[30] Yiming Li, Tongqing Zhai, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor attack in the physical world. In *ICLR Workshop*, 2021. 2

[31] Zhifeng Li, Dihong Gong, Qiang Li, Dacheng Tao, and Xuelong Li. Mutual component analysis for heterogeneous face recognition. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 7(3):1–23, 2016. 1

[32] Zhifeng Li, Dihong Gong, Xuelong Li, and Dacheng Tao. Learning compact feature descriptor and adaptive matching framework for face recognition. *IEEE Transactions on Image Processing*, 24(9):2736–2745, 2015. 1

[33] Zhifeng Li, Dihong Gong, Yu Qiao, and Dacheng Tao. Common feature discriminant analysis for matching infrared face images to optical face images. *IEEE transactions on image processing*, 23(6):2436–2445, 2014. 1

[34] 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*, pages 273–294. Springer, 2018. 2, 5

[35] Wei Liu, Zhifeng Li, and Xiaoou Tang. Spatio-temporal embedding for statistical face recognition from video. In *ECCV*, 2006. 1

[36] Yunfei Liu, Xingjun Ma, James Bailey, and Feng Lu. Reflection backdoor: A natural backdoor attack on deep neural networks. In *ECCV*, 2020. 2, 5

[37] Yue Ma, Yali Wang, Yue Wu, Ziyu Lyu, Siran Chen, Xiu Li, and Yu Qiao. Visual knowledge graph for human action reasoning in videos. In *ACM MM*, 2022. 1

[38] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018. 8

[39] Anh Nguyen and Anh Tran. Wanet–imperceptible warping-based backdoor attack. In *ICLR*, 2021. 2, 5

[40] Tuan Anh Nguyen and Anh Tran. Input-aware dynamic backdoor attack. In *NeurIPS*, 2020. 2, 5

[41] Youngtaek Oh, Dong-Jin Kim, and In So Kweon. Distribution-aware semantics-oriented pseudo-label for imbalanced semi-supervised learning. In *CVPR*, 2022. 3

[42] Xiangyu Qi, Tinghao Xie, Yiming Li, Saeed Mahloujifar, and Prateek Mittal. Revisiting the assumption of latent separability for backdoor defenses. In *ICLR*, 2023. 1

[43] 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. 1

[44] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The german traffic sign recognition benchmark: a multi-class classification competition. In *IJCNN*, 2011. 5

[45] Xiaoou Tang and Zhifeng Li. Video based face recognition using multiple classifiers. In *Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004. Proceedings.*, pages 345–349. IEEE, 2004. 1

[46] Guanhong Tao, Guangyu Shen, Yingqi Liu, Shengwei An, Qiuling Xu, Shiqing Ma, Pan Li, and Xiangyu Zhang. Better trigger inversion optimization in backdoor scanning. In *CVPR*, 2022. 2

[47] Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. In *NeurIPS*, 2018. 2

[48] Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Clean-label backdoor attacks. 2018. 2, 3, 5, 8

[49] 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 *SP*, 2019. 2

[50] Haotao Wang, Junyuan Hong, Aston Zhang, Jiayu Zhou, and Zhangyang Wang. Trap and replace: Defending backdoor attacks by trapping them into an easy-to-replace subnetwork. In *NeurIPS*, 2022. 2

[51] Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. Symmetric cross entropy for robust learning with noisy labels. In *ICCV*, 2019. 2, 4

[52] Maurice Weber, Xiaojun Xu, Bojan Karlaš, Ce Zhang, and Bo Li. Rab: Provable robustness against backdoor attacks. In *IEEE SP*, 2022. 2

[53] Emily Wenger, Josephine Passananti, Arjun Nitin Bhagoji, Yuanshun Yao, Haitao Zheng, and Ben Y Zhao. Backdoor attacks against deep learning systems in the physical world. In *CVPR*, 2021. 2

[54] Dongxian Wu and Yisen Wang. Adversarial neuron pruning purifies backdoored deep models. In *NeurIPS*, 2021. 2

[55] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. In *NeurIPS*, 2020. 7

[56] Yi Zeng, Si Chen, Won Park, Z Morley Mao, Ming Jin, and Ruoxi Jia. Adversarial unlearning of backdoors via implicit hypergradient. In *ICLR*, 2022. 2

[57] Tong Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *ICML*, 2004. 8

[58] Pu Zhao, Pin-Yu Chen, Payel Das, Karthikeyan Natesan Ramamurthy, and Xue Lin. Bridging mode connectivity in loss landscapes and adversarial robustness. In *ICLR*, 2020. 2

[59] Shihao Zhao, Xingjun Ma, Xiang Zheng, James Bailey, Jingjing Chen, and Yu-Gang Jiang. Clean-label backdoor attacks on video recognition models. In *CVPR*, 2020. 2

[60] Zhendong Zhao, Xiaojun Chen, Yuexin Xuan, Ye Dong, Dakui Wang, and Kaitai Liang. Defeat: Deep hidden feature backdoor attacks by imperceptible perturbation and latent representation constraints. In *CVPR*, 2022. 2

[61] Mingkai Zheng, Shan You, Lang Huang, Fei Wang, Chen Qian, and Chang Xu. Simmatch: Semi-supervised learning with similarity matching. In *CVPR*, 2022. 3

[62] Runkai Zheng, Rongjun Tang, Jianze Li, and Li Liu. Data-free backdoor removal based on channel lipschitzness. In *ECCV*, 2022. 2