Pattern Recognition 139 (2023) 109512

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Not All Samples Are Born Equal: Towards Effective Clean-Label Backdoor Attacks

Yinghua Gao[a,1], Yiming Li[a,1,*], Linghui Zhu[a,b,1], Dongxian Wu[c], Yong Jiang[a,b], Shu-Tao Xia[a,b]

[a] Tsinghua Shenzhen International Graduate School, Tsinghua University, China
[b] Research Center of Artificial Intelligence, Peng Cheng Laboratory, China
[c] Department of Complexity Science and Engineering, The University of Tokyo, Japan

ARTICLE INFO

Article history:
Received 31 October 2022
Revised 3 February 2023
Accepted 4 March 2023
Available online 10 March 2023

Keywords:
Backdoor attack
Clean-label attack
Sample selection
Trustworthy ML
AI Security
Deep learning

ABSTRACT

Recent studies demonstrated that deep neural networks (DNNs) are vulnerable to backdoor attacks. The attacked model behaves normally on benign samples, while its predictions are misled whenever adversary-specified trigger patterns appear. Currently, clean-label backdoor attacks are usually regarded as the most stealthy methods in which adversaries can only poison samples from the target class without modifying their labels. However, these attacks can hardly succeed. In this paper, we reveal that the difficulty of clean-label attacks mainly lies in the antagonistic effects of 'robust features' related to the target class contained in poisoned samples. Specifically, robust features tend to be easily learned by victim models and thus undermine the learning of trigger patterns. Based on these understandings, we propose a simple yet effective plug-in method to enhance clean-label backdoor attacks by poisoning 'hard' instead of random samples. We adopt three classical difficulty metrics as examples to implement our method. We demonstrate that our method can consistently improve vanilla attacks, based on extensive experiments on benchmark datasets.

1. Introduction

Deep learning, especially deep neural networks (DNNs), has been successfully and widely developed in many applications [1–3] for its high effectiveness and efficiency [4–6]. Currently, the training of DNNs is data-driven, requiring a large number of samples and computational resources. Accordingly, researchers and developers usually need to exploit third-party resources (*e.g.*, training samples from the Internet or company, cloud computing platforms, and pre-trained backbones) to reduce the training costs in practice.

However, recent studies revealed that using third-party training resources could introduce a new threatening security risk, which was called backdoor attacks [7]. In general, backdoor adversaries intend to embed the hidden backdoor, *i.e.*, a latent connection between the adversary-specified trigger pattern and the target label, to victim DNNs by maliciously manipulating their training process. For example, the adversaries can randomly select some samples from the original benign training dataset and modify their images by adding pre-defined trigger patterns and re-assign their label as the adversary-specified target label. Those generated poisoned samples associated with the remaining benign ones forms the poisoned training set, which is released to the victims for training their models. Currently, there are many different types of backdoor attacks, such as invisible attacks [8–10], sample-specific attacks [11–13], and clean-label attacks [14–16]. Among all different types of methods, clean-label backdoor attacks are usually regarded as the most stealthy yet difficult ones for they can bypass human inspection while hard to succeed. Different from attacks with poisoned labels, the clean-label backdoor adversaries can only poison samples from the target class without modifying their labels.

In this paper, we further explore clean-label backdoor attacks. We demonstrate that the difficulty of clean-label attacks mainly lies in the antagonistic effects of 'robust features' (*i.e.*, semantic features related to the target label) contained in the original samples. For example, the robust features could refer to the 'hairy ears' and 'black noses' for images in the dog class. These features are easier to be learned by DNNs and hinder models from building a connection between the trigger pattern and the target label. Besides, we summarize that Turner et al. [14] reduced the ability of robust features by performing adversarial attacks on selected samples (based on a DNN with adversarial training [17]) before adding the trigger pattern, while Zhao et al. [15] enhanced trigger ability by using targeted universal adversarial perturbation [18] as the trigger pattern to alleviate this problem. In particular, we notice that both aforementioned methods treated all data equally since

* Corresponding author.
E-mail addresses: li-ym18@mails.tsinghua.edu.cn (Y. Li), xiast@sz.tsinghua.edu.cn (S.-T. Xia).
[1] indicates equal contribution

https://doi.org/10.1016/j.patcog.2023.109512
0031-3203/

they randomly select samples for poisoning. In other words, both of them have a latent assumption that the robust features contained in all samples have the same ability. It raises two intriguing questions: **1)** *Are all training samples born equal?* **2)** *If not, how can we exploit this characteristic to design more effective clean-label backdoor attacks?*

The answer to the first question is in the negative. In this paper, we reveal that DNNs perform differently in terms of their learning ability for different training samples. Some samples are easier to be learned while others are not. Based on these findings, we propose a simple yet effective plug-in method to further enhance the effectiveness of clean-label backdoor attacks by poisoning 'hard' instead of random samples. The robust features contained in these hard samples are naturally less effective. Specifically, we adopt three classical difficulty metrics, including **1)** loss value, **2)** gradient norm, and **3)** forgetting event, as examples for selecting hard samples. We demonstrate that all of them are effective in improving clean-label attacks. Besides, we also demonstrate that our method (under all difficulty metrics) has high transferability, where poisoned samples generated based on one model structure are also promising in attacking others. Moreover, we also show that the enhanced clean-label attacks with our plug-in method keep the resistance to potential defenses of their original version, *i.e.*, our method does not decrease attack stealthiness.

In conclusion, our main contributions can be summarized in three-fold. **1)** We reveal that the difficulty of clean-label backdoor attacks is mostly due to the antagonistic effects of 'robust features' and verify that DNNs have different learning abilities for different training samples. **2)** We revisit the paradigm of existing clean-label backdoor attacks and propose a new complementary paradigm by considering the different learning difficulties of samples. **3)** We empirically verify the effectiveness and the poisoning transferability of our method on benchmark datasets and discuss its intrinsic mechanism.

2. Related works

In this paper, we focus on backdoor attacks and defenses in image classification. Methods in other tasks are out of our scope.

2.1. Backdoor attacks

Backdoor attacks is an emerging yet severe threat to deep neural networks (DNNs). Different from adversarial attacks [19–21] that target the inference process, backdoor threats may happen when the training and deployment processes of DNNs are not fully controlled [7]. Currently, existing attacks can be divided into three main categories, including **1)** poison-only attacks, **2)** training-controlled attacks, and **3)** non-poisoning-based attacks.

Specifically, poison-only backdoor attacks [12,22,23] required that the adversaries can only modify the training dataset, whereas having neither the information nor the ability to modify other training components (*e.g.*, training loss, training schedule, and model structure); Training-controlled attacks [9,24,25] allowed adversaries having full control over the training process including the training data and the training algorithm; Different from previous approaches, non-poisoning-based attacks [26–28] focused mainly on the deployment rather than the training phase. These methods embedded hidden backdoors by direct weight modification or introducing malicious DNN modules.

In this paper, we mainly focus on the poison-only backdoor attack, which is the most classical setting having the widest threatening scenarios. In particular, we separate existing poison-only methods into two sub-categories, as follows:

Poison-only Backdoor Attacks with Poisoned Labels. In these attacks, backdoor adversaries re-assigned the labels of poisoned samples that are different from the ground-truth ones of their benign version. For example, a cat-like poisoned image may be labeled as the 'dog'. It is currently the most widespread attack paradigm for its simplicity and effectiveness. BadNets [22] is the first poison-only backdoor attack with poisoned labels. Specifically, it randomly selected some benign samples and modified their images by stamping on an adversary-specified trigger pattern. Those generated poisoned samples associated with the remaining benign ones forms the poisoned training set used for training victim models. Following its attack paradigm, many other methods were also proposed with different trigger designs [8,12,13,29–31].

Poison-only Backdoor Attacks with Clean Labels. Recently, Turner et al. [14] argued that having poisoned images resemble their benign version is not stealthy enough if the attack is still with poisoned labels. Dataset users could still identify these attacks by examining the consistency between the image and its label. For example, if a dog-like image is labeled as a 'cat', users can treat it as a malicious sample even if the image seems to be innocent. Accordingly, they proposed label-consistent backdoor attack (LC) by poisoning samples only from the target class. In particular, they first leveraged adversarial perturbations [17], generated based on an adversarially robust DNN under adversarial training [17], to modify selected images before adding triggers. After that, Zhao et al. [15] proposed to exploit targeted universal adversarial perturbation (TUAP) [18] as the trigger pattern. Deep neural networks are more likely to learn the TUAP since it is a 'stronger' feature. Recently, there were also a few other clean-label backdoor attacks. However, they either required controlling the training process [32] or were designed for other purposes having limited threats [16]. How to design more effective poison-only clean-label backdoor attacks is still an important open question.

2.2. Backdoor defenses

Currently, there were also some backdoor defenses to alleviate backdoor threats. In general, we can divide existing methods into three main categories, including **1)** backdoor elimination, **2)** image pre-processing, and **3)** poison detection. Specifically, model elimination [33–35] aimed to remove hidden backdoors or prevent their creation. For example, defenders can exploit model pruning [33] and knowledge distillation [34] to remove embedded backdoors while introducing randomness or decoupling the training process to prevent their creation; Image pre-processing [36–38] transformed all input testing images before feeding them into the deployed model for predictions. These transformations (*e.g.*, spatial transformations and image reconstruction) are the feasible methods to change or even remove trigger patterns even if defenders have no information about potential attacks, leading to the deactivation of embedded model backdoors; Except for directly reducing backdoor threats, poison detection [39–41] identified whether a suspicious third-party object (*i.e.*, sample or model) is malicious. For example, STRIP [40] superimposed different images on the suspicious image and treated it as the malicious one if the predictions of generated images have low randomness measured by the entropy.

2.3. Sample selection in deep learning

Sample selection has been a long-standing research topic in deep learning, where users select and exploit specific instead of random or all samples for training [42]. In most cases, sample selection can remove outliers and find informative samples, facilitating the effectiveness of downstream tasks.

Currently, there are many different metrics for sample selection. Arguably, **1)** loss value, **2)** gradient norm and **3)** forgetting event, are the three most classical and widely used metrics. Specifically,

the loss value of training samples was usually used in noisy label learning [43–45] and curriculum learning [46–48]. For example, Jiang et al. [43] adopted it to distinguish correctly labeled samples from the corrupted training set by filtering samples with the smallest loss values; Kumar et al. [46] proposed to exploit loss value to measure the difficulty of training samples, based on which to train DNNs from easier data to harder data. Gradient norm has also been widely used for sample selection [49,50], inspired by the understanding that sample gradients can reflect their contributions and significance. Users can also exploit it to measure sample difficulty during the training process since a large gradient norm generally indicates that the sample has not been well learned. Recently, Toneva et al. [51] revealed that samples learned in one epoch may be forgotten in the next. This phenomenon was called the forgetting event. In general, the larger the number of forgetting events, the harder and the more important the sample for training the model. A well-performed model can be obtained by discarding the samples which have never undergone forgetting.

In this paper, we focus on how to select the most suitable samples for poisoning to improve the success rate of backdoor attacks rather than improving benign accuracy. We believe it can provide further insights into sample selection, which has been ignored in previous works.

3. Revisiting clean-label attacks and the training dynamic

In this section, we revisit representative clean-label backdoor attacks and the training dynamic of samples. We will show that robust features (*i.e.*, semantic features related to the target label) contained in poisoned samples are in competition to trigger patterns during the training process of DNNs and the robust features in some samples may be less effective. These findings motivate our method, which will be designed and illustrated in the next section.

3.1. Antagonistic effects of robust features in clean-label backdoor attacks

In clean-label attacks, both robust and trigger-related features are likely to be exploited to predict the target label during model training. We hereby investigate how robust features influence these attacks.

Settings. Recall that the adversaries of LC added adversarial perturbations with maximum perturbation size ϵ, based on a DNN trained with adversarial training, before implanting trigger patterns. In particular, recent studies [52–54] revealed that DNNs obtained via adversarial training mostly adopt 'robust features' for classification. Accordingly, the larger the ϵ, the weaker the robust features. Different from LC, TUAP exploited the targeted universal adversarial perturbation with maximum perturbation size ϵ as the trigger pattern. As such, the larger the ϵ, the 'stronger' the trigger-related features. In this part, we vary the value of ϵ from 4 pixels to 32 pixels to demonstrate the antagonistic effects of robust features in clean-label backdoor attacks. Specifically, we conduct experiments on the CIFAR-10 dataset [55] with ResNet-18 [56]. The trigger pattern of the label-consistent attack is all-white patches located in four corners. Besides, we run each experiment with five random seeds and report the mean and standard deviation to reduce the side effects of randomness.

Results. As shown in Fig. 1–2, the attack success rate (ASR) significantly increases with the increase of the maximum perturbation size ϵ in both cases. In other words, reducing the effects of robust features or increasing that of trigger-related ones can improve the effectiveness of clean-label backdoor attacks. These phenomena indicate that *robust features and trigger-related features are antagonistic in the training process of DNNs*.

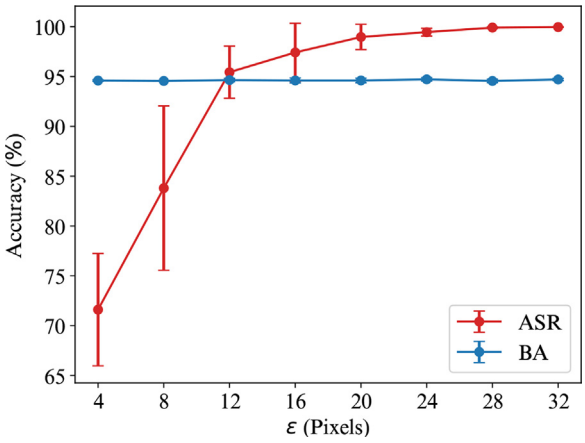


Fig. 1. The benign accuracy (BA) and attack success rate (ASR) of LC attack w.r.t different ϵ values.

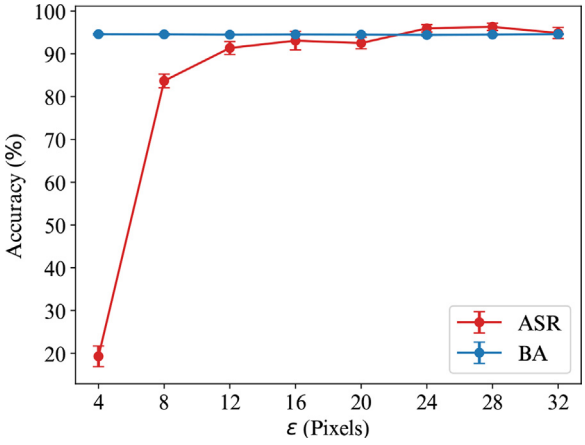


Fig. 2. The benign accuracy (BA) and attack success rate (ASR) of TUAP attack w.r.t different ϵ values.

3.2. Not all samples are born equal

In this section, we revisit the training dynamic of samples to verify that some samples are easier to be learned while others are not.

Settings. We train a ResNet-18 model on the benign instead of the poisoned CIFAR-10 dataset. We calculate the learning indicator of each sample in each epoch during the training process measuring whether it is correctly classified by the current model, based on which to obtain forgetting events of each data. We call there is a forgetting event if the sample was correctly classified in the previous epoch whereas being misclassified in the next epoch.

Results. As shown in Fig. 3, there is a massive difference in how easy they are to be learned even for samples from the same class. Specifically, although most of them are relatively simple (#Forgetting Events<5), there still exists a significant portion of 'hard' samples that are very difficult to be learned. *The robust features contained in these hard samples are naturally less effective and therefore have less side-effects in learning trigger patterns*. In particular, we visualize the training dynamic of a hard sample with a large number of forgetting events. As shown in Fig. 4, it is repeatedly remembered yet repeatedly forgotten before the training process is converged.

4. The proposed method

Section 3.1 reveals that the robust features contained in poisoned samples hinder the learning of trigger patterns, while

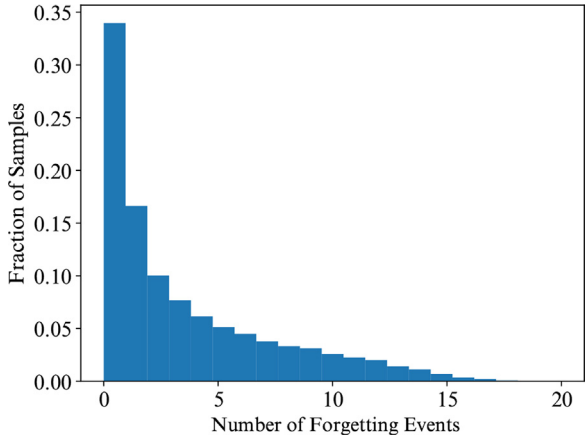


Fig. 3. The histogram of forgetting events for samples in the 'airplane' class.

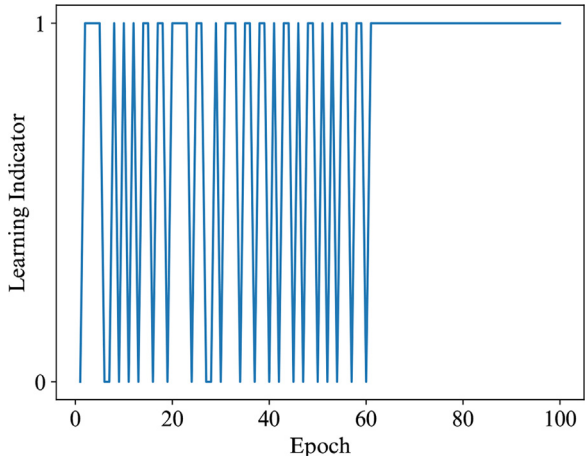


Fig. 4. The learning behavior of a' hard' training sample over 100 epochs.

Section 3.2 shows that the effectiveness of robust features varies across samples. Motivated by these findings, we propose a simple yet effective plug-in method to improve the effectiveness of clean-label backdoor attacks by poisoning 'hard' instead of random samples.

4.1. Preliminaries

Notations. We denote by $\mathcal{T} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ the (unmodified) benign training set, where $\boldsymbol{x}_i \in \mathbb{R}^d$ and $y_i \in [K] = \{1, 2, \cdots, K\}$ are the image and its label, respectively. Let y_t denotes the target label determined by the adversary and $\mathcal{T}_t$ ($\mathcal{T}_t \subset \mathcal{T}$) contains all samples whose label is y_t. Let $\mathcal{S}$ represents a small subset of $\mathcal{T}_t$: $\mathcal{S} \subset \mathcal{T}_t$ and $|\mathcal{S}| \ll |\mathcal{T}_t|$. $f_{\boldsymbol{w}} : \mathbb{R}^d \to [0,1]^K$ indicates the classifier that outputs the probability vector parameterized by $\boldsymbol{w}$, where d is the dimension of images. $\mathcal{L}$ is the loss function (*e.g.*, cross-entropy loss). $G_{\boldsymbol{\theta}} : \mathbb{R}^d \to \mathbb{R}^d$ denotes the poisoned input generator parameterized by $\boldsymbol{\theta}$. $\boldsymbol{m} \in \{0,1\}^d$ is a binary mask, $\boldsymbol{t} \in \mathbb{R}^d$ is the trigger pattern, and $\odot$ represents the Hadamard product.

Threat Model. We consider the *poison-only* backdoor attacks: the adversary has access to the original training set $\mathcal{T}$ and is allowed to manipulate a small portion of training data to inject the pre-defined trigger. Specifically, the attacked training set $\mathcal{T}'$ is composed of two parts: the poisoned subset $\mathcal{T}_p = \{(G_{\boldsymbol{\theta}}(\boldsymbol{x}_i), y_i) | (\boldsymbol{x}_i, y_t) \in \mathcal{S}\}$ and the remaining benign subset $\mathcal{T}_b = \mathcal{T} - \mathcal{S}$. However, the adversary has no information and the ability to modify other training components, such as loss functions and model structures. The adversary will release the attacked dataset $\mathcal{T}'$ to the victim for training their models. The backdoor (*i.e.*a latent connection between the trigger pattern and the target label) will be created during the training process. WLOG, let $p \triangleq 100 \cdot \frac{|\mathcal{S}|}{|\mathcal{T}_t|}$ denotes the adversary-specified *poisoning rate*. It is the percentage of the number of poisoned training samples (*i.e.*, $|\mathcal{S}|$) over that of samples whose ground-truth label is the target label (*i.e.*, $|\mathcal{T}_t|$). In general, poison-only attacks require minimal capacities and could happen in many real-world scenarios [7], such as using third-party training samples, platforms, and pre-trained models.

4.2. Clean-label backdoor attacks with hard samples

The core technique in our method is how to select samples (*i.e.*, $\mathcal{S}$) for poisoning. Specifically, instead of poisoning randomly selected samples, we advocate poisoning only hard samples. In this section, we introduce three classical difficulty metrics to implement our method.

Loss Value. Given a benign model $f_{\boldsymbol{w}}$ (trained on the benign training set $\mathcal{T}$), we denote the loss value of sample $(\boldsymbol{x}_i, y_i)$ as $\mathcal{L}(f_{\boldsymbol{w}}(\boldsymbol{x}_i), y_i)$ and data with the greatest $p\%$ values are chosen for poisoning:

$$\mathcal{S} = \arg\max_{\mathcal{S} \subset \mathcal{T}_t} \sum_{(\boldsymbol{x}_i, y_i) \in \mathcal{S}} \mathcal{L}(f_{\boldsymbol{w}}(\boldsymbol{x}_i), y_i). \tag{1}$$

Gradient Norm. Given a benign model $f_{\boldsymbol{w}}$ (trained on the benign training set $\mathcal{T}$), we denote the (ℓ_2-) gradient norm of sample $(\boldsymbol{x}_i, y_i)$ as $\| \nabla_{\boldsymbol{w}} \mathcal{L}(f_{\boldsymbol{w}}(\boldsymbol{x}_i), y_i)\|_2$ and samples with the greatest $p\%$ values are chosen:

$$\mathcal{S} = \arg\max_{\mathcal{S} \subset \mathcal{T}_t} \sum_{(\boldsymbol{x}_i, y_i) \in \mathcal{S}} \| \nabla_{\boldsymbol{w}} \mathcal{L}(f_{\boldsymbol{w}}(\boldsymbol{x}_i), y_i)\|_2. \tag{2}$$

Forgetting Event. We count the number of occurrences of a forgetting event and denote the statistics of sample $\boldsymbol{x}_i$ as E_i during the training process. Those samples with the greatest $p\%$ forgetting statistics are chosen for trigger injection:

$$\mathcal{S} = \arg\max_{\mathcal{S} \subset \mathcal{T}_t} \sum_{(\boldsymbol{x}_i, y_i) \in \mathcal{S}} E_i. \tag{3}$$

After $\mathcal{S}$ is determined, we poison these samples with the poisoned input generator $G_{\boldsymbol{\theta}}$, which is specified by backdoor adversaries. For example, in BadNets-type attacks, $G_{\boldsymbol{\theta}}(\boldsymbol{x}) = (1 - \boldsymbol{m}) \odot \boldsymbol{x} + \boldsymbol{m} \odot \boldsymbol{t}$. The attacked dateset $\mathcal{T}'$ will be used by victims to train their model $g_{\boldsymbol{w}}$ based on the standard training process

$$\min_{\boldsymbol{w}} \sum_{(\boldsymbol{x}, y) \in \mathcal{T}'} \mathcal{L}(g_{\boldsymbol{w}}(\boldsymbol{x}), y). \tag{4}$$

In particular, g may be different from f which is used by the adversary. As we will show in Section 5.4.1, our method is still effective in these cases, since the generated poisoned samples have promising transferability.

In the inference process, given any 'unseen' sample $(\boldsymbol{x}, y)$ where $y \neq y_t$, the adversary can activate backdoors to maliciously manipulate the prediction of $g_{\boldsymbol{w}}$ to the target class using $G_{\boldsymbol{\theta}}(\boldsymbol{x})$ instead of $\boldsymbol{x}$.

5. Experiments

5.1. Main settings

Dataset and Model. We conduct experiments on two benchmark datasets, including CIFAR-10 [55] and ImageNet [57], with ResNet-18 [56]. Due to the limitation of computational resources, we randomly select a subset from the original ImageNet dataset containing 50 classes where each class consists of 500 images for training and 50 images for testing.

(a) Benign (b) BadNets-C (c) Blended-C (d) LC (e) TUAP

Fig. 5. The example of samples involved in different attacks on the CIFAR-10 dataset.

(a) Benign (b) BadNets-C (c) Blended-C (d) LC (e) TUAP

Fig. 6. The example of samples involved in different attacks on the ImageNet dataset.

Baseline Selection. We compare attacks with our plug-in methods to their standard version with the random selection strategy (dubbed 'vanilla'). Specifically, we evaluate all methods under two representative clean-label backdoor attacks, including LC [14] and TUAP [15]. We also provide the clean-label variant of two classical poisoned-label attacks, including BadNets (dubbed 'BadNets-C') [22] and attack with the blended strategy (dubbed 'Blended-C'), by poisoning samples only from the target class.

Attack Setup on CIFAR-10. For BadNets, we use a 3×3 checkerboard trigger. For Blended, we choose a Hello-Kitty trigger and blend this trigger with the original images. The transparency parameter is set to 0.2. For LC, we pre-train an adversarially robust model, based on which to add the adversarial perturbation (via PGD [17] under ℓ_∞ norm) to the original images with the perturbation budget set as 8 pixels before introducing trigger patterns. Then we perturb selected images by a four-corner all-white trigger. For TUAP, we first generate the targeted universal adversarial perturbation with the budget set as 4 pixels and exploit it as additive trigger patterns. For all these attacks, we poison 10% samples from the target class (*i.e.*, 1% over all samples) and choose the first class (*i.e.*, 'airplane') as the target class. The example of poisoned samples generated by different attacks is shown in Fig. 5.

Attack Setup on ImageNet. For BadNets, we use a 20×20 white trigger patch. For Blended, we choose a random noise trigger and blend this trigger with the original images with transparency $\alpha = 0.2$. For LC, we pre-train an adversarially robust model, based on which to add the adversarial perturbation to the original images with the perturbation budget set as 8 pixels before introducing trigger patterns. After that, we add a four-corner trigger to the perturbed images. For TUAP, The perturbation budget is also set as 8 pixels. The target class is assigned as '0' for all attacks and we poison 50% samples from the target class (*i.e.*, 1% over all samples). The example of poisoned samples generated by different attacks is shown in Fig. 6.

Evaluation Metric. We evaluate all backdoor attacks with two class metrics, including BA and ASR. BA is the prediction accuracy on all testing samples, while ASR is the accuracy on their poisoned version. The ultimate goal of adversaries is ensuring attack effectiveness (high ASR) while maintaining model standard functionality (high BA).

5.2. Main results

As shown in Table 1–2, poisoning only hard samples instead of random sample consistently improve all clean-label attacks by a significant margin, no matter which difficulty metric is adopted. For example, on the CIFAR-10 dataset, using our methods can improve ASR by more than 20% compared to the vanilla version of all attacks in almost all cases. The improvements are even more than 30% in some cases on CIFAR-10, such as using forgetting event for BadNets-C. In particular, the benign accuracy of attacks with our selection strategies is on par with (*i.e.*, the decrease $< 1\%$) that of using standard random selection. These results verify the effectiveness of our methods.

Besides, we notice that the ASRs of our methods may have a relatively large difference across datasets. We speculate that it is mostly because different datasets have different abilities of robust features and trigger-related features. We will further explore its mechanism in our future work.

5.3. Ablation study

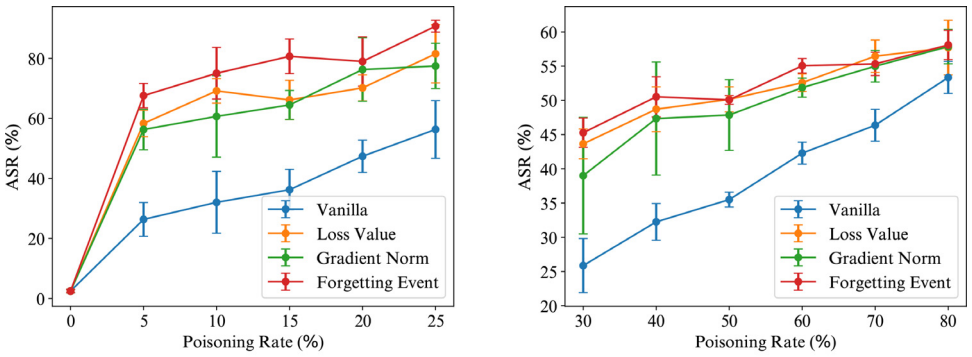
In this section, we discuss the effects of key hyper-parameters involved in our methods. For simplicity, we adopt BadNets-C on

Table 1
The main results (%) on the CIFAR-10 dataset.

Method↓	Attack→ Metric↓	BadNets-C	Blended-C	LC	TUAP
Vanilla	BA	94.57 ± 0.16	94.53 ± 0.15	94.55 ± 0.04	94.60 ± 0.19
	ASR	32.02 ± 10.32	44.67 ± 3.00	83.80 ± 8.24	19.29 ± 2.42
Loss Value (Ours)	BA	94.48 ± 0.17	94.43 ± 0.22	94.50 ± 0.20	94.42 ± 0.13
	ASR	69.15 ± 4.07	64.85 ± 6.33	95.41 ± 1.90	39.68 ± 4.82
Gradient Norm (Ours)	BA	94.38 ± 0.10	94.47 ± 0.17	94.60 ± 0.12	94.50 ± 0.16
	ASR	60.66 ± 13.57	65.21 ± 5.62	94.09 ± 2.65	37.63 ± 5.53
Forgetting Event (Ours)	BA	94.43 ± 0.21	94.28 ± 0.13	94.47 ± 0.16	94.32 ± 0.18
	ASR	75.04 ± 8.60	73.85 ± 2.78	98.21 ± 0.85	48.42 ± 2.91

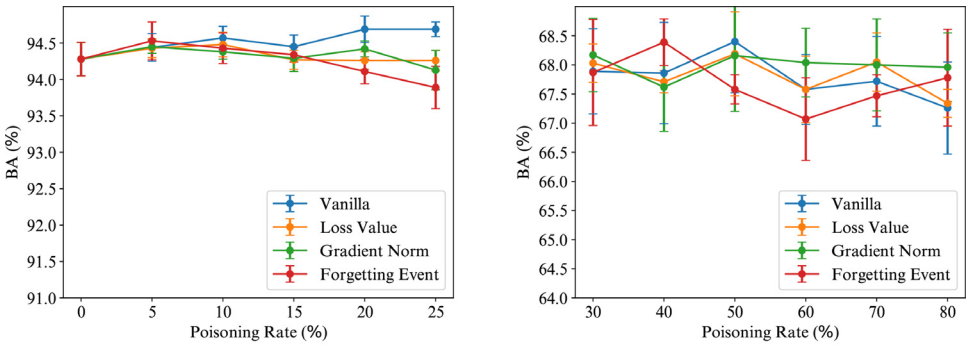
Table 2
The main results (%) on the ImageNet dataset.

Method↓	Attack→ Metric↓	BadNets-C	Blended-C	LC	TUAP
Vanilla	BA	68.40 ± 0.88	68.11 ± 0.51	67.68 ± 0.31	67.67 ± 0.58
	ASR	35.50 ± 1.08	14.18 ± 1.87	49.29 ± 1.43	36.16 ± 2.26
Loss Value (Ours)	BA	68.19 ± 0.72	68.02 ± 0.47	67.80 ± 0.54	67.72 ± 0.57
	ASR	50.15 ± 1.83	25.28 ± 3.78	60.96 ± 3.61	48.61 ± 2.10
Gradient Norm (Ours)	BA	68.16 ± 0.96	68.42 ± 0.94	67.99 ± 0.79	67.83 ± 0.96
	ASR	47.86 ± 5.16	25.30 ± 5.05	59.82 ± 5.37	45.66 ± 6.39
Forgetting Event (Ours)	BA	67.58 ± 0.25	68.61 ± 0.92	67.89 ± 0.87	67.86 ± 0.77
	ASR	50.07 ± 0.64	30.36 ± 3.68	62.42 ± 2.49	50.53 ± 2.85



(a) CIFAR-10 (b) ImageNet

Fig. 7. The attack success rate (ASR) of BadNets-C with respect to the poisoning rate on the target class on CIFAR-10 and ImageNet datasets.



(a) CIFAR-10 (b) ImageNet

Fig. 8. The benign accuracy (BA) of BadNets-C with respect to the poisoning rate on the target class on CIFAR-10 and ImageNet datasets.

the CIFAR-10 dataset as an example for our discussions. Unless otherwise specified, all settings are the same as those used in Section 5.2.

Effects of the Poisoning Rate. We evaluate the BA and ASR of methods with different poisoning rates in the target class. As shown in Fig. 7, the ASR increases with the increase of the poisoning rate. We notice that the improvements between our methods and the vanilla attack may decrease with its increase, especially when the poisoning rate is relatively large. It is mostly because samples selected by the random strategy also contain many hard samples when the poisoning rate is large. Besides, as shown in Fig. 8, increasing the poisoning rate will also reduce BA, although the effects are relatively minor.

Effect of Trigger Patterns. Besides the checkerboard trigger used in Section 5.2, we evaluate whether our methods are still effective when using other trigger patterns. Specifically, we adopt three representative trigger patterns, including the all-white patch, the all-black patch, and the random noise patch. The visualization of trigger patterns and the experimental results are in Table 3. Although its performance may have some fluctuations across different trigger patterns, our methods can consistently and significantly improve the performance of vanilla attacks (as shown in Table 3).

Effects of the Target Label. We also evaluate whether our methods are still effective when using different target labels. Specifically, we select four additional target labels, including '2' (*i.e.*, bird), '4' (*i.e.*, deer), '6' (*i.e.*, frog), and '8' (*i.e.*, ship). As shown in Table 4, although the performance may have some differences across different target labels, our methods can still consistently and significantly improve the performance of vanilla attacks.

5.4. Discussion

5.4.1. The poisoning transferability across model structures

In previous experiments, the target model (*i.e.*, the model adopted by victim dataset users) is the same as that of the source model (*i.e.*, the model used by backdoor adversaries to generate the attacked training dataset). However, in practical scenarios, the adversaries have no information about the training process,

Table 3
The performance (%) of BadNets-C with different trigger patterns on CIFAR-10.

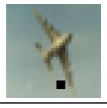
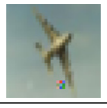
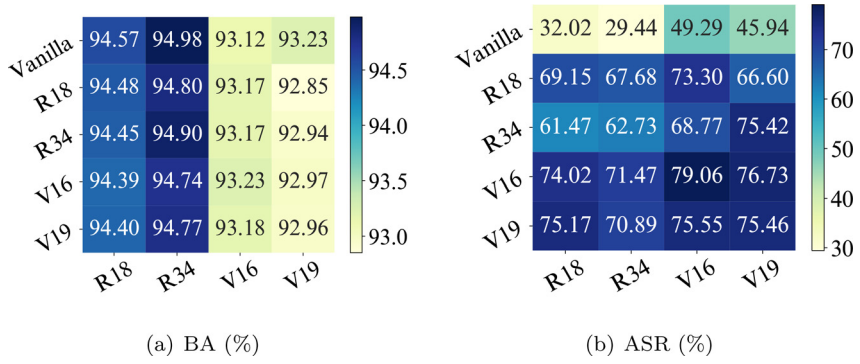
Method↓	Pattern→ Metric↓			
Vanilla	BA	94.57 ± 0.04	94.50 ± 0.07	94.60 ± 0.09
	ASR	16.87 ± 2.86	20.47 ± 5.81	55.13 ± 6.92
Loss Value (Ours)	BA	94.41 ± 0.13	94.50 ± 0.18	94.54 ± 0.13
	ASR	45.34 ± 3.43	53.93 ± 6.62	88.01 ± 8.10
Gradient Norm (Ours)	BA	94.55 ± 0.18	94.49 ± 0.14	94.49 ± 0.14
	ASR	42.85 ± 10.12	53.81 ± 5.86	88.22 ± 4.56
Forgetting Event (Ours)	BA	94.31 ± 0.20	94.56 ± 0.16	94.31 ± 0.15
	ASR	61.88 ± 7.33	69.24 ± 6.97	97.54 ± 0.36

Table 4
The performance (%) of BadNets-C with different target labels (y_t) on CIFAR-10.

Method↓	y_t → Metric↓	2	4	6	8
Vanilla	BA	94.52 ± 0.25	94.41 ± 0.15	94.58 ± 0.13	94.52 ± 0.13
	ASR	66.55 ± 9.46	50.57 ± 11.46	28.26 ± 5.29	30.00 ± 3.83
Loss Value (Ours)	BA	94.37 ± 0.13	94.52 ± 0.06	94.62 ± 0.17	94.61 ± 0.20
	ASR	87.62 ± 6.37	78.16 ± 5.25	72.33 ± 14.19	82.24 ± 7.14
Gradient Norm (Ours)	BA	94.36 ± 0.26	94.53 ± 0.14	94.51 ± 0.16	94.57 ± 0.22
	ASR	82.09 ± 5.42	76.02 ± 11.79	68.57 ± 24.20	75.44 ± 7.62
Forgetting Event (Ours)	BA	94.27 ± 0.14	94.29 ± 0.03	94.38 ± 0.14	94.37 ± 0.18
	ASR	89.56 ± 4.15	85.00 ± 4.72	88.74 ± 5.11	83.46 ± 4.89



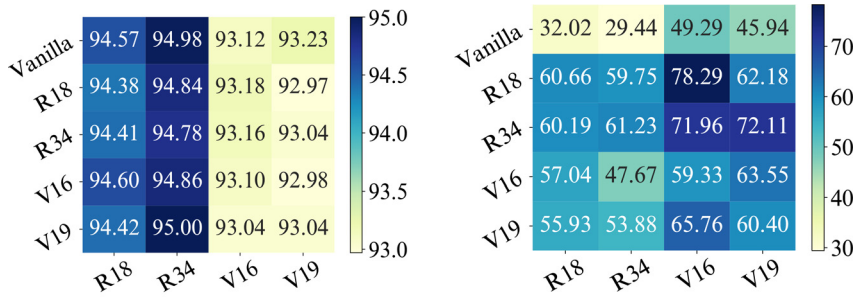
(a) BA (%) (b) ASR (%)

Fig. 9. The poisoning transferability of our method with loss value under BadNets-C on the CIFAR-10 dataset. **Row**: source model. **Column**: target model. The first row (*i.e.*, 'Vanilla') indicates the results of poisoning random samples with the target model.

including the model structure (see Section 4.2). In this part, we discuss whether our method can transfer across different model structures.

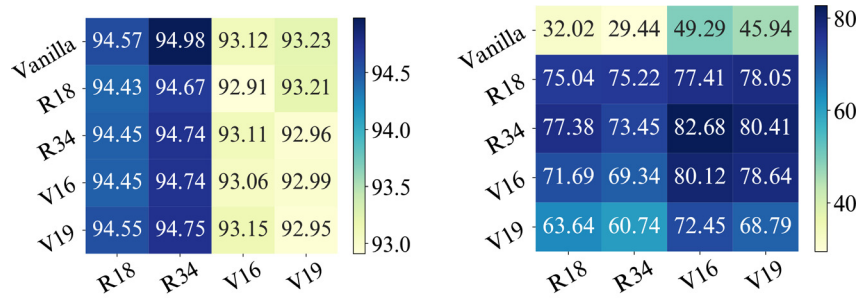
Settings. We take BadNets-C on the CIFAR-10 dataset as an example for our discussions. Specifically, we adopt four representative model structures, including ResNet-18, ResNet-34, VGG-16, and VGG-19, for our evaluation. Besides, we also provide the results of the vanilla attack (using the random selection strategy) for reference. Except for the model structure, all other settings are the same as those used in Section 5.2.

Results. As shown in Fig. 9–11, our methods have high poisoning transferability in terms of high ASRs across different model structures. Although the performance may have some fluctuations, our methods are consistently and significantly better than the



(a) BA (%) (b) ASR (%)

Fig. 10. The poisoning transferability of our method with gradient norm under BadNets-C on the CIFAR-10 dataset. **Row**: source model. **Column**: target model. The first row (*i.e.*, 'Vanilla') indicates the results of poisoning random samples with the target model.



(a) BA (%) (b) ASR (%)

Fig. 11. The poisoning transferability of our method with forgetting event under BadNets-C on the CIFAR-10 dataset. **Row**: source model. **Column**: target model. The first row (*i.e.*, 'Vanilla') indicates the results of poisoning random samples with the target model.

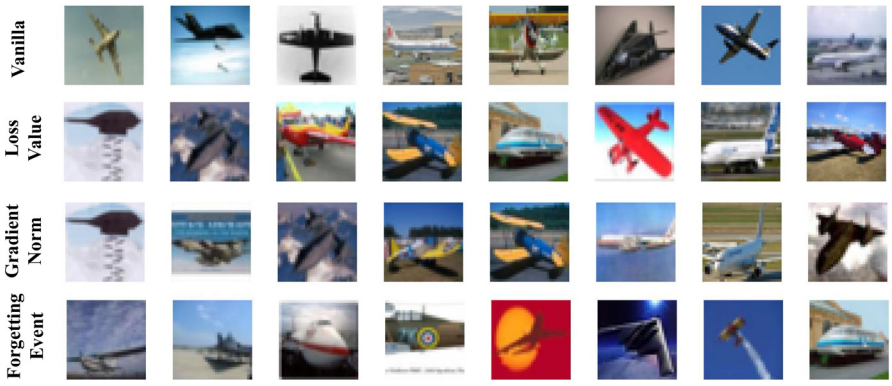


Fig. 12. Samples selected by different methods on the CIFAR-10 dataset.

vanilla attack, even when the target model is significantly different from the source one (*e.g.*, ResNet-34 to VGG-16). In particular, we notice that using the same source and target model may not always lead to the best performance. This phenomenon may be related to the learning ability of different models. We will further explore its mechanism and discuss how to optimize the selection across models in our future work.

5.4.2. Why our methods are effective?

In this section, we further discuss why our methods are effective. Similar to previous experiments, we adopt BadNets-C on the CIFAR-10 dataset as an example for our discussions. All settings are the same as those in Section 5.1.

Firstly, we visualize some samples selected by different methods. As shown in Fig. 12, samples chosen by the vanilla attack are generally easier for human inspection. Specifically, the airplane contained in samples selected by our methods is either smaller, at a strange angle, or of a rare type. These results verify that our methods can find hard samples, whose robust features are naturally weaker.

Secondly, we visualize the feature representation of samples generated by models attacked by different methods via t-SNE [58]. As shown in Fig. 13, the poisoned samples generated by our methods form a more compact and separate cluster, compared to those generated by the vanilla attack. This phenomenon also partly explains our effectiveness, since it is easier for the remaining fully-connected layers to assign the target label to a compact and separate cluster.

5.4.3. Applying our methods to poisoned-label backdoor attacks

In the previous experiments, we apply our sample selection strategies to clean-label attacks. In this section, we discuss whether it can also be adopted to improve poisoned-label backdoor attacks.

Table 5
The results of applying our methods to poisoned-label attacks on CIFAR-10.

Method↓	Metric↓, Attack→	BadNets	Blended
Vanilla	BA (%)	94.43 ± 0.19	94.31 ± 0.28
	ASR (%)	89.88 ± 11.28	20.68 ± 1.40
Loss Value (Ours)	BA (%)	94.36 ± 0.17	94.32 ± 0.27
	ASR (%)	82.53 ± 11.54	20.59 ± 3.19
Gradient Norm (Ours)	BA (%)	94.42 ± 0.26	94.46 ± 0.27
	ASR (%)	76.84 ± 17.56	17.36 ± 5.45
Forgetting Event (Ours)	BA (%)	94.42 ± 0.29	94.43 ± 0.16
	ASR (%)	78.03 ± 6.99	23.07 ± 2.17

Settings. Similar to previous sections, we adopt BadNets and Blended on the CIFAR-10 dataset as an example for our discussions. Under the poisoned-label setting, the poisoned samples are selected from the whole training set rather than the target class. All other settings are the same as those in Section 5.1.

Results. As shown in Table 5, our method has no benefit in improving the attack success rate. In most cases, using our methods may even decrease the ASR. This failure is mostly because the difficulty metrics are calculated according to the ground-truth label instead of the target label. In particular, it is not feasible to estimate the difficulties of each sample according to the target label since reassigning all labels to the targeted one will significantly reduce benign accuracy. We will further discuss how to generalize our methods to improve poisoned-label attacks in our future works.

5.4.4. The resistance to potential defenses

In this section, we demonstrate that our methods will not decrease the resistance of the vanilla attack to potential backdoor

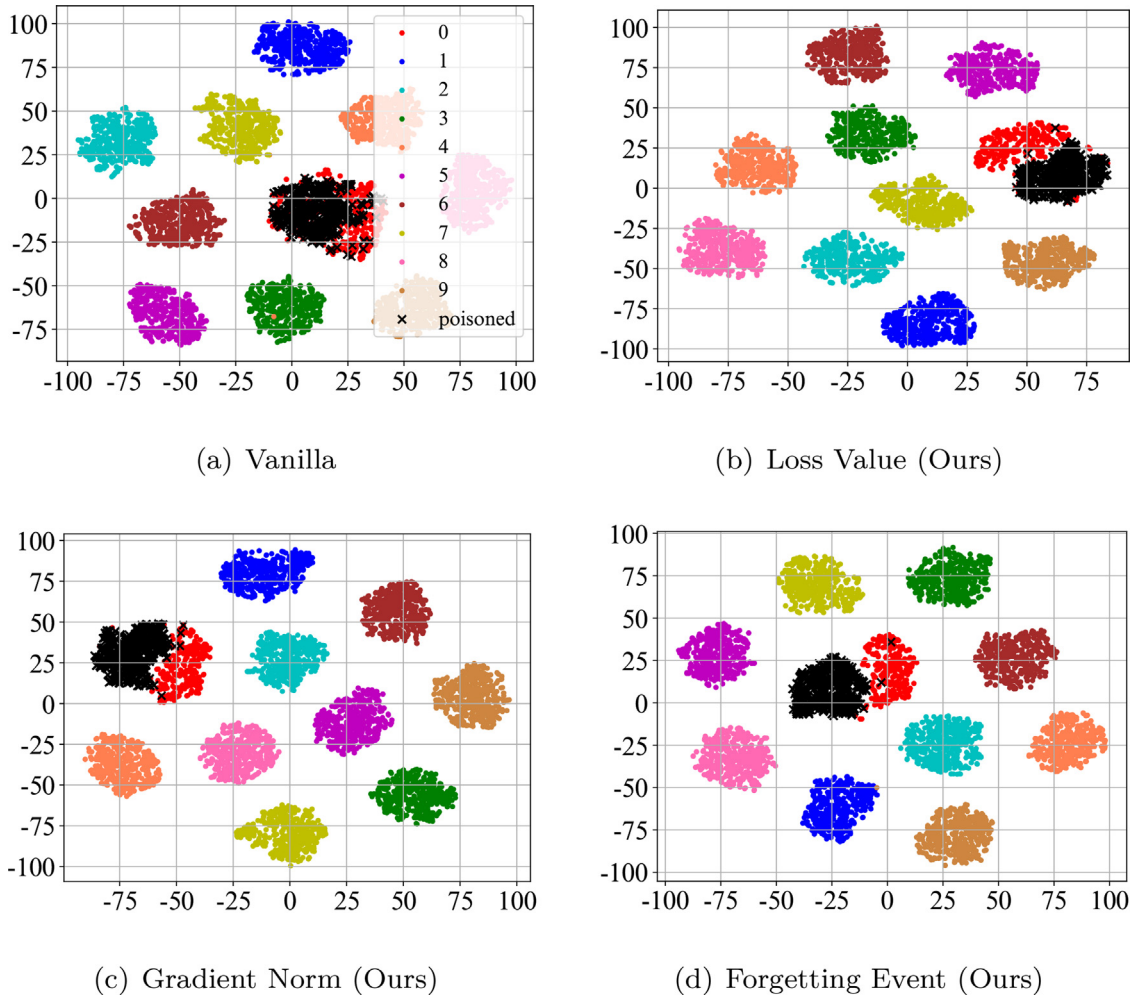


Fig. 13. The t-SNE of feature representations of samples generated by BadNets-C with different sample selection methods on the CIFAR-10 dataset.

Table 6
The resistance of BadNets-C to fine-pruning on the CIFAR-10 dataset.

Method↓	Pruning Rate→ Metric↓	0%	10%	20%	30%
Vanilla	BA (%)	94.57 ± 0.16	93.59 ± 0.55	92.36 ± 0.52	90.71 ± 0.90
	ASR (%)	32.02 ± 10.32	26.29 ± 6.49	26.68 ± 10.16	23.96 ± 7.92
Loss Value (Ours)	BA (%)	94.48 ± 0.17	93.64 ± 0.55	92.76 ± 0.40	90.21 ± 0.66
	ASR (%)	69.15 ± 4.07	46.88 ± 18.23	38.37 ± 13.90	27.47 ± 15.98
Gradient Norm (Ours)	BA (%)	94.38 ± 0.10	93.80 ± 0.29	92.42 ± 0.43	89.95 ± 1.20
	ASR (%)	60.66 ± 13.57	58.67 ± 18.49	45.73 ± 12.70	24.60 ± 14.96
Forgetting Event (Ours)	BA (%)	94.43 ± 0.21	93.67 ± 0.31	92.22 ± 0.53	90.33 ± 0.52
	ASR (%)	75.04 ± 8.60	48.34 ± 34.63	51.07 ± 21.86	36.33 ± 24.00

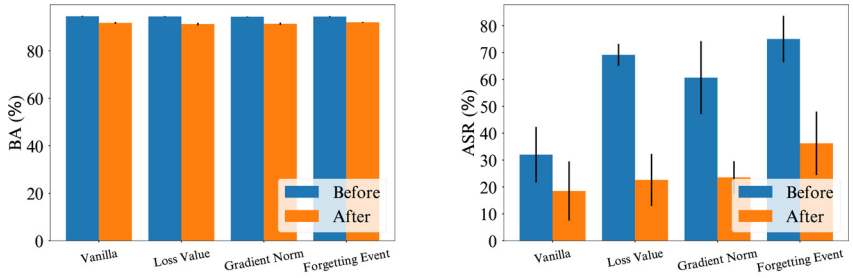
defenses. Unless otherwise specified, we adopt BadNets-C on the CIFAR-10 dataset for the discussions.

The Resistance to Backdoor Elimination. We evaluate all methods under two representative defenses, including fine-pruning (FP) [33] and neural attention distillation (NAD) [34]. For FP, the pruning rate ranges from 0% to 30%. For NAD, the attention distillation hyper-parameter β is set to 500. The initial learning rate is 0.1 and we tune attacked models 20 epochs in total. We implement these defenses based on the open-sourced toolbox `BackdoorBox`[2]. As shown in Table 6 and Fig. 14, BadNets-C with our sample selection methods is still consistently and significantly better than its original version, although the ASR may degrade sharply after the defenses.

The Resistance to Image Pre-processing. We evaluate all methods under two representative pre-processing methods, including Auto-Encoder [36] and ShrinkPad [38]. We implement these methods based on `BackdoorBox` [59] with their default settings. As shown in Table 7, our methods can not make the vanilla more resistant to these defenses, although they can significantly improve the performance without defenses. How to design more robust clean-label backdoor attacks is still an important open question that is out of the scope of this paper. We will further explore it in our future works.

The Resistance to Poison Detection. We evaluate all methods under two representative detection methods, including STRIP [40] and SentiNet [39]. STRIP adopted prediction randomness among all perturbed samples measured by Shannon entropy to detect poisoned samples, while SentiNet exploited Grad-CAM [60] to detect trigger regions. As shown in Fig. 15–16, the detection results

[2] https://github.com/THUYimingLi/BackdoorBox



(a) Benign Accuracy (b) Attack Success Rate

Fig. 14. The resistance of BadNets-C to NAD on the CIFAR-10 dataset.

Table 7
The resistance of BadNets-C to Auto-Encoder and ShrinkPad on CIFAR-10.

Defense↓	Method→ Metric↓	Vanilla	Loss Value	Gradient Norm	Forgetting Event
No Defense	BA (%)	94.57 ± 0.16	94.48 ± 0.17	94.38 ± 0.10	94.43 ± 0.21
	ASR (%)	32.02 ± 10.32	69.15 ± 4.07	60.66 ± 13.57	75.04 ± 8.60
Auto-Encoder	BA (%)	88.09 ± 0.61	88.33 ± 0.16	88.37 ± 0.13	87.76 ± 0.15
	ASR (%)	2.27 ± 0.36	2.35 ± 0.39	2.09 ± 0.42	1.86 ± 0.28
ShrinkPad	BA (%)	91.91 ± 0.22	91.51 ± 0.22	91.77 ± 0.42	91.75 ± 0.15
	ASR (%)	4.83 ± 0.35	4.89 ± 1.14	4.55 ± 0.88	4.59 ± 0.41

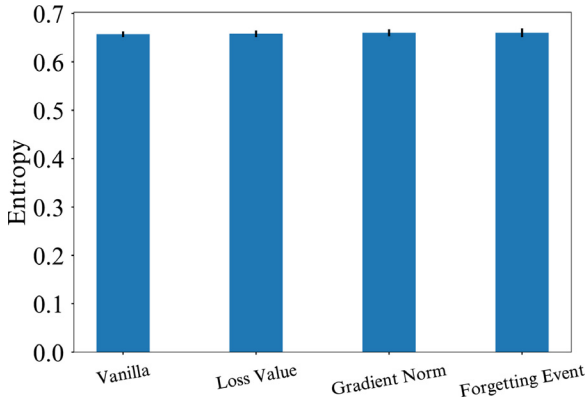


Fig. 15. The resistance of BadNets-C to STRIP on the CIFAR-10 dataset.

among all our methods and the vanilla attack are similar. These results verify that our methods do not decrease the resistance to defenses.

5.4.5. Clean-label backdoor attacks in the physical world

In this section, we verify that attacks with our methods are still effective in the physical world if their vanilla versions are effective.

Settings. We adopt four models obtained in Section 5.2 trained on the poisoned ImageNet dataset under the BadNets-C with an all-white trigger patch and different sample selection strategies for our discussions. We also include the model trained on the benign ImageNet dataset for reference. Specifically, we take pictures of a cola bottle with and without the trigger pattern via a mobile device and feed them into all five models (*i.e.*, four backdoored models and one benign model). The benign and attacked images are shown in Fig. 17.

Results. All five models can correctly classify the benign image. In particular, the benign model can also correctly classify the

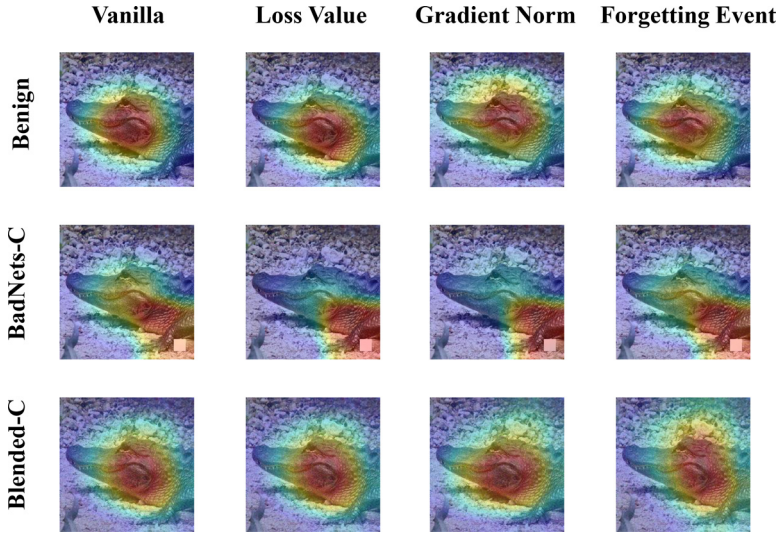


Fig. 16. The Grad-CAM of samples generated by different methods on ImageNet.

(a) Benign Image (b) Attacked Image

Fig. 17. The benign image and attacked image of a 'Bottle' object taken by a mobile device. Both benign and backdoored models can correctly classify the benign image as the 'Bottom', while only attacked models can predict the attacked images as the target class (*i.e.*, 'iPod').

attacked image, although there is a white patch stamped on the bottle. However, the predictions of this image generated by four attacked models are all the 'iPod', which is the target label. These results indicate that attacks using our methods are still effective in the real physical world if their vanilla versions are effective.

6. Conclusion

In this paper, we revisited the clean-label backdoor attacks. We revealed that their difficulties mainly lie in the antagonistic effects of robust features related to the target class in the poisoned samples. We also noticed that existing clean-label attacks had a false latent assumption that robust features contained in all samples have the same ability. Based on these findings, we proposed a simple yet effective plug-in method by poisoning 'hard' instead of random samples. We exploited three classical difficulty metrics as examples to measure sample difficulty. Extensive experiments on benchmark datasets were conducted, which verified the effectiveness of our method.

However, we also noticed that our method is not feasible to improve poisoned-label attacks and could not increase the resistance of vanilla attacks to potential backdoor defenses. We will further alleviate these problems in our future work.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

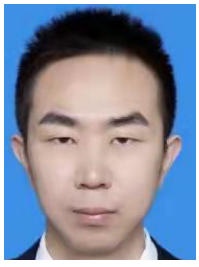
Acknowledgement

This work is supported in part by the National Natural Science Foundation of China under Grant 62171248, Shenzhen Science and Technology Program (JCYJ20220818101012025), the PCNL KEY project (PCL2021A07), and Research Center for Computer Network (Shenzhen) Ministry of Education.

References

[1] Z. Li, D. Gong, Y. Qiao, D. Tao, Common feature discriminant analysis for matching infrared face images to optical face images, IEEE Trans. Image Process. 23 (6) (2014) 2436–2445.
[2] L. Song, H. Wang, Z.J. Wang, Decoupling multi-task causality for improved skin lesion segmentation and classification, Pattern Recognit. 133 (2023) 108995.
[3] Y. Zhou, Z. Huang, X. Yang, M. Ang, T.K. Ng, Gcm: efficient video recognition with glance and combine module, Pattern Recognit. 133 (2023) 108970.
[4] H. Qin, R. Gong, X. Liu, X. Bai, J. Song, N. Sebe, Binary neural networks: a survey, Pattern Recognit. 105 (2020) 107281.
[5] H. Qiu, D. Gong, Z. Li, W. Liu, D. Tao, End2end occluded face recognition by masking corrupted features, IEEE Trans. Pattern Anal. Mach. Intell. (2021).
[6] H. Qin, Y. Ding, M. Zhang, Q. Yan, A. Liu, Q. Dang, Z. Liu, X. Liu, Bibert: accurate fully binarized bert, ICLR, 2022.
[7] Y. Li, Y. Jiang, Z. Li, S.-T. Xia, Backdoor learning: a survey, IEEE Trans. Neural Netw. Learn. Syst. (2022).
[8] X. Chen, C. Liu, B. Li, K. Lu, D. Song, Targeted backdoor attacks on deep learning systems using data poisoning, arXiv preprint arXiv:1712.05526 (2017).
[9] Y. Zeng, W. Park, Z.M. Mao, R. Jia, Rethinking the backdoor attacks' triggers: a frequency perspective, ICCV, 2021.
[10] J. Hayase, S. Oh, Few-shot backdoor attacks via neural tangent kernels, ICLR, 2023.
[11] T.A. Nguyen, A. Tran, Input-aware dynamic backdoor attack, NeurIPS, 2020.
[12] Y. Li, Y. Li, B. Wu, L. Li, R. He, S. Lyu, Invisible backdoor attack with sample-specific triggers, ICCV, 2021.
[13] J. Zhang, C. Dongdong, Q. Huang, J. Liao, W. Zhang, H. Feng, G. Hua, N. Yu, Poison ink: robust and invisible backdoor attack, IEEE Trans. Image Process. 31 (2022) 5691–5705.
[14] A. Turner, D. Tsipras, A. Madry, Label-consistent backdoor attacks, arXiv preprint arXiv:1912.02771 (2019).
[15] S. Zhao, X. Ma, X. Zheng, J. Bailey, J. Chen, Y.-G. Jiang, Clean-label backdoor attacks on video recognition models, CVPR, 2020.
[16] Y. Li, Y. Bai, Y. Jiang, Y. Yang, S.-T. Xia, B. Li, Untargeted backdoor watermark: towards harmless and stealthy dataset copyright protection, NeurIPS, 2022.
[17] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks, ICLR, 2018.
[18] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, P. Frossard, Universal adversarial perturbations, CVPR, 2017.
[19] A. Liu, X. Liu, J. Fan, Y. Ma, A. Zhang, H. Xie, D. Tao, Perceptual-sensitive gan for generating adversarial patches, AAAI, 2019.
[20] A. Liu, J. Wang, X. Liu, b. Cao, C. Zhang, H. Yu, Bias-based universal adversarial patch attack for automatic check-out, ECCV, 2020.
[21] J. Wang, A. Liu, Z. Yin, S. Liu, S. Tang, X. Liu, Dual attention suppression attack: Generate adversarial camouflage in physical world, CVPR, 2021.
[22] T. Gu, K. Liu, B. Dolan-Gavitt, S. Garg, Badnets: evaluating backdooring attacks on deep neural networks, IEEE Access 7 (2019) 47230–47244.
[23] X. Qi, T. Xie, Y. Li, S. Mahloujifar, P. Mittal, Revisiting the assumption of latent separability for backdoor defenses, ICLR, 2023.
[24] E. Bagdasaryan, V. Shmatikov, Blind backdoors in deep learning models, USENIX Security, 2021.
[25] Y. Li, H. Zhong, X. Ma, Y. Jiang, S.-T. Xia, Few-shot backdoor attacks on visual object tracking, ICLR, 2022.
[26] R. Tang, M. Du, N. Liu, F. Yang, X. Hu, An embarrassingly simple approach for trojan attack in deep neural networks, KDD, 2020.
[27] X. Qi, T. Xie, R. Pan, J. Zhu, Y. Yang, K. Bu, Towards practical deployment-stage backdoor attack on deep neural networks, CVPR, 2022.
[28] J. Bai, K. Gao, D. Gong, S.-T. Xia, Z. Li, W. Liu, Hardly perceptible trojan attack against neural networks with bit flips, ECCV, 2022.
[29] J. Lin, L. Xu, Y. Liu, X. Zhang, Composite backdoor attack for deep neural network by mixing existing benign features, CCS, 2020.
[30] A. Nguyen, A. Tran, Wanet–imperceptible warping-based backdoor attack, ICLR, 2021.
[31] M. Xue, C. He, J. Wang, W. Liu, One-to-n & n-to-one: two advanced backdoor attacks against deep learning models, IEEE Trans. Dependable Secure Comput. 19 (03) (2022) 1562–1578.
[32] A. Saha, A. Subramanya, H. Pirsiavash, Hidden trigger backdoor attacks, AAAI, 2020.
[33] K. Liu, B. Dolan-Gavitt, S. Garg, Fine-pruning: defending against backdooring attacks on deep neural networks, RAID, 2018.
[34] Y. Li, X. Lyu, N. Koren, L. Lyu, B. Li, X. Ma, Neural attention distillation: erasing backdoor triggers from deep neural networks, ICLR, 2021.
[35] K. Huang, Y. Li, B. Wu, Z. Qin, K. Ren, Backdoor defense via decoupling the training process, ICLR, 2022.
[36] Y. Liu, Y. Xie, A. Srivastava, Neural trojans, ICCD, 2017.
[37] H. Qiu, Y. Zeng, S. Guo, T. Zhang, M. Qiu, B. Thuraisingham, Deepsweep: an evaluation framework for mitigating dnn backdoor attacks using data augmentation, Asia CCS, 2021.
[38] Y. Li, T. Zhai, Y. Jiang, Z. Li, S.-T. Xia, Backdoor attack in the physical world, ICLR Workshop, 2021.
[39] E. Chou, F. Tramer, G. Pellegrino, Sentinet: detecting localized universal attack against deep learning systems, IEEE S&P Workshop, 2020.
[40] Y. Gao, Y. Kim, B.G. Doan, Z. Zhang, G. Zhang, S. Nepal, D.C. Ranasinghe, H. Kim, Design and evaluation of a multi-domain trojan detection method on deep neural networks, IEEE Trans. Dependable Secure Comput. 19 (4) (2022) 2349–2364.
[41] J. Guo, Y. Li, X. Chen, H. Guo, L. Sun, C. Liu, Scale-up: an efficient black-box input-level backdoor detection via analyzing scaled prediction consistency, ICLR, 2023.
[42] C. Guo, B. Zhao, Y. Bai, Deepcore: a comprehensive library for coreset selection in deep learning, arXiv preprint arXiv:2204.08499 (2022).
[43] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, L. Fei-Fei, Mentornet: learning data-driven curriculum for very deep neural networks on corrupted labels, ICML, 2018.
[44] B. Han, G. Niu, X. Yu, Q. Yao, M. Xu, I. Tsang, M. Sugiyama, Sigua: forgetting may make learning with noisy labels more robust, ICML, 2020.

[45] C. Tan, J. Xia, L. Wu, S.Z. Li, Co-learning: learning from noisy labels with self–supervision, ACM MM, 2021.
[46] M. Kumar, B. Packer, D. Koller, Self-paced learning for latent variable models, NeurIPS, 2010.
[47] F. Ma, D. Meng, Q. Xie, Z. Li, X. Dong, Self-paced co-training, ICML, 2017.
[48] X. Wang, Y. Chen, W. Zhu, A survey on curriculum learning, IEEE Trans. Pattern Anal. Mach. Intell. (2021).
[49] P. Zhao, T. Zhang, Stochastic optimization with importance sampling for regularized loss minimization, ICML, 2015.
[50] A. Katharopoulos, F. Fleuret, Not all samples are created equal: deep learning with importance sampling, ICML, 2018.
[51] M. Toneva, A. Sordoni, R.T.d. Combes, A. Trischler, Y. Bengio, G.J. Gordon, An empirical study of example forgetting during deep neural network learning, ICLR, 2019.
[52] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, A. Madry, Adversarial examples are not bugs, they are features, NeurIPS, 2019.
[53] M. Terzi, A. Achille, M. Maggipinto, G.A. Susto, Adversarial training reduces information and improves transferability, AAAI, 2021.
[54] Z. Allen-Zhu, Y. Li, Feature purification: how adversarial training performs robust deep learning, FOCS, 2022.
[55] A. Krizhevsky, Learning Multiple Layers of Features from Tiny Images, Master's thesis, University of Tront (2009).
[56] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, CVPR, 2016.
[57] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: a large-scale hierarchical image database, CVPR, 2009.
[58] L. Van der Maaten, G. Hinton, Visualizing data using t-sne, J. Mach. Learn. Res. 9 (11) (2008).
[59] Y. Li, M. Ya, Y. Bai, Y. Jiang, S.-T. Xia, Backdoorbox: a python toolbox for backdoor learning, ICLR Workshop (2023).
[60] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-cam: visual explanations from deep networks via gradient-based localization, ICCV, 2017.



Yinghua Gao received his BS degree from the Department of Mathematics at Nankai University. He is currently a Ph.D. student in the Department of Computer Science and Technology at Tsinghua University. His research interests are primarily in trustworthy machine learning.

Yiming Li is currently a Ph.D. candidate in Computer Science and Technology from Tsinghua Shenzhen International Graduate School, Tsinghua University, China. Before that, he received his B.S. degree with honors in Mathematics and Applied Mathematics from Ningbo University, China, in 2018. His research interests are in the domain of Trustworthy ML, especially backdoor learning, adversarial learning, data privacy, and copyright protection in deep learning. His research has been published in multiple top-tier conferences and journals, such as ICLR, NeurIPS, ICCV, AAAI, Pattern Recognition, and IEEE TNNLS. He served as the senior program committee member of AAAI, the program committee member of ICLR, NeurIPS, ICML, etc., and the reviewer of IEEE TPAMI, IEEE TIFS, IEEE TDSC, Pattern Recognition, etc.

Linghui Zhu received the B.S. degree in Computer Science and Technology from Nankai University, Tianjin, China, in 2020. She is currently pursuing the master degree in Tsinghua Shenzhen International Graduate School, Tsinghua University. Her research interests are in the domain of data security and federated learning.

Dr. Dongxian Wu is currently a Postdoctoral Researcher at the University of Tokyo. He received his Bachelor degree in Microelectronics at Xidian University in 2016 and received his Ph.D. degree in Computer Science and Technology at Tsinghua University in 2021. He focuses on trustworthy machine learning, especially adversarial learning and data security.

Dr. Yong Jiang received his M.S. and Ph.D. degrees in computer science from Tsinghua University, China, in 1998 and 2002, respectively. Since 2002, he has been with the Tsinghua Shenzhen International Graduate School of Tsinghua University, Guangdong, China, where he is currently a full professor. His research interests include computer vision, machine learning, Internet architecture and its protocols, IP routing technology, etc. He has received several best paper awards from top-tier conferences and his research has been published in multiple top-tier journals and conferences, including IEEE ToC, IEEE TMM, IEEE TSP, CVPR, ICLR, etc.

Dr. Shu-Tao Xia received the B. S. degree in mathematics and the Ph.D. degree in applied mathematics from Nankai University, Tianjin, China, in 1992 and 1997, respectively. Since January 2004, he has been with the Graduate School at Shenzhen of Tsinghua University, Guangdong, China. He is now a full professor there. From March 1997 to April 1999, he was with the research group of information theory, Department of Mathematics, Nankai University, Tianjin, China. From September 1997 to March 1998 and from August to September 1998, he visited the Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong. His current research interests include coding and information theory, networking, machine learning, and deep learning. His researches have been published in multiple top-tier conferences and journals, including IEEE TPAMI, IEEE TIP, CVPR, ICLR, etc.