Towards Understanding How Self-training Tolerates Data Backdoor Poisoning

Soumyadeep Pal[1,*], Ren Wang[2], Yuguang Yao[3] and Sijia Liu[4]

[1]University of Alberta
[2]Illinois Institute of Technology
[3]Michigan State University
[4]Michigan State University

Abstract

Recent studies on backdoor attacks in model training have shown that polluting a small portion of training data is sufficient to produce incorrect manipulated predictions on poisoned test-time data while maintaining high clean accuracy in downstream tasks. The stealthiness of backdoor attacks has imposed tremendous defense challenges in today's machine learning paradigm. In this paper, we explore the potential of self-training via additional unlabeled data for mitigating backdoor attacks. We begin by making a pilot study to show that vanilla self-training is *not* effective in backdoor mitigation. Spurred by that, we propose to defend the backdoor attacks by leveraging strong but proper data augmentations in the self-training pseudo-labeling stage. We find that the new self-training regime help in defending against backdoor attacks to a great extent. Its effectiveness is demonstrated through experiments for different backdoor triggers on CIFAR-10 and a combination of CIFAR-10 with an additional unlabeled 500K TinyImages dataset. Finally, we explore the direction of combining self-supervised representation learning with self-training for further improvement in backdoor defense.

Keywords

backdoor attack, data poisoning, self-training, deep learning

1. Introduction

Deep neural networks (DNNs), key components of deep learning, have prompted a technological revolution in artificial intelligence through various applications in computer vision [1, 2, 3] and other realms [4, 5]. Due to the ever-growing capacity of DNNs, the models are capable of learning better and more accurately during the training phase. This can sometimes lead to DNNs being brittle - (1) Well-crafted imperceptible perturbations on test images can cause the model to misclassify images during the inference stage (known as evasion attack) [6]. (2) Another attack called data poisoning attack [7] can occur first during training by manipulating the training data by the introduction of toxic artifacts. These are memorized by the model and are carried on to the inference stage. Attack type (2) is the major focus of this work.

Recent DNNs are extremely data-hungry - they are often trained using data from anonymous or unverified sources from the internet. This makes it particularly convenient for adversaries to manipulate datasets, leading to various kinds of stealthy data poisoning attacks, thus posing a real threat to deep learning security [8]. One of such data poisoning attacks is the backdoor attack (also known as Trojan attack) [9, 7], where a fraction of the training data is corrupted by the addition of a trigger. In this paper, we focus on defense against such backdoor attacks.

In many applications of deep learning, there is the availability of large quantities of unlabeled data - labeling is often cumbersome due to time and resources. Hence, semi-supervised learning [10] has been a growing area of research, which aims to leverage such unlabeled data to improve the performance of DNNs. Self-training is one such popular paradigm, which has been proven to perform really well in large data settings [11]. In this context, we aim to address the following question:

(Q) How does self-training relate to robustness against backdoor attacks ?

Self-training has been recently shown to have some capability of using diverse feature priors during training [12] and help alleviate spurious correlations under certain sets of assumptions [13]. This inspires our study towards understanding if this paradigm may be helpful in backdoor defense.

To the best of our knowledge, the most relevant work to ours is [14]. In [14], self-supervised learning and symmetric cross entropy loss [15] is used to separate the data into probable clean and poisoned samples. Then semi-supervised learning (MixUp [16]) is performed with the

Safe AI '23: Feb 13-14, 2023 | Washington, D.C., US @AAAI-23
*Corresponding author.
soumyade@ualberta.ca (S. Pal); rwang74@iit.edu (R. Wang); yaoyugua@msu.edu (Y. Yao); liusiji5@msu.edu (S. Liu)
https://wangren09.github.io/ (R. Wang); https://lsjxjtu.github.io/ (S. Liu)
© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
CEUR Workshop Proceedings (CEUR-WS.org)

CEUR-WS.org/Vol-3381/31.pdf

filtered clean samples as labeled data and the rest as unlabeled data (by removing their labels). However, this method is not able to answer our question (Q). Different from [14], we do not aim to filter out clean samples using heuristic detection method. Though we use self-training as a semi-supervised learning algorithm for mitigating backdoor, we aim to get insights on how self-training as a sole learning paradigm can help in backdoor mitigation. We summarise our **contributions** as follows:

- We show that self-training can mitigate backdoor using additional clean unlabeled data
- We propose that self-training combined carefully with data augmentation has the capacity to defend against backdoor to a certain extent, even when the unlabeled data is poisoned.
- Stronger defense is possible if stochastic data augmentation schemes (like in SimCLR [17]) are employed with self-training

2. Related Works

2.1. Backdoor Attacks

Backdoor attack is one of the emerging fields of research in data poisoning while training neural networks. We focus on two types of trigger-driven backdoor attacks - poisoned label attacks and clean label backdoor attacks.

The poisoned label attacks constitute of poisoning the training dataset by injecting a trigger in a small portion of the dataset and mislabeling them to a target class. This was fundamentally demonstrated in BadNets [18] which used a rectangular patch and stamped it on an area of an image. Subsequently, more sophisticated triggers have been developed [19, 20, 21, 22].

Another type of backdoor attack constitutes the clean label backdoor attack [23, 24]. The images belonging to the target class are adversarially perturbed away from their true class and then injected with the trigger. Training with such images establishes the correlation between the trigger and the target class. This type of attack is more stealthy because the labels of the target class are consistent with the ground truth labels.

In this paper, we consider both types - the basic poisoned label and the clean label backdoor attack for our experiments.

2.2. Backdoor Defense

Due to the emerging threat of backdoor attacks, several kinds of defenses have been proposed. These roughly belong to the following categories : (1) *Input preprocessing* [25, 26, 27, 28]: This kind of defense introduces a preprocessing module with the intent of damaging the trigger pattern before passing it into the DNN. (2) *Detection based defense* [29, 30, 31, 32, 33]: The aim here is to detect the presence of possible malicious samples or backdoored models. Then the method either denies the use of such suspicious object or filters the suspicious input samples for re-training. (3) *Erasure based or model reconstruction based defense* [34, 35, 36, 37]: This type of defense aims to erase the effect of triggers from an already backdoored model such that it performs well in both clean samples and in the presence of triggers. (4) *Trigger synthesis* [38, 39, 40, 41]: Here the trigger is potentially detected and synthesized in the first step and then the effect of such a trigger is suppressed. (5) *Poison suppression defense* [42, 43]: This kind of defense tries to suppress the effectiveness of hidden triggers in the input samples during training, thus preventing the model from learning any correlation with the trigger.

In this paper, we aim to suppress the poison using data augmentation and erase its effect from a trained poisoned model by self-training. Thus, our work falls within the scope of poison suppression and erasure-based defense.

2.3. Self-training

Self-training is a form of semi-supervised learning [10] which attempts to leverage unlabeled data to improve classification performance in the limited data regime. Different types of semi-supervised learning paradigms have been explored such as consistency training [44, 45, 46, 47] and pseudo-labeling [48, 49, 11].

In self-training, a good teacher model is initially trained using the labeled data. This model is used to generate pseudolabels for the unlabeled data which are then used to train a student model. This same process is repeated iteratively.

The main rationale behind this method is that a trained teacher model would provide better predictions on unlabeled data than pure chance. Because of the uncertainty of the correctness of predicted pseudolabels, a confidence-based example selection scheme [50] is often employed. Here, a fraction of pseudolabels for which the teacher model assigns the highest probability is used to train the student model. This is repeated with increasing fractions of unlabeled data till completion.

In recent studies, self-training has been shown to have some capacity to incorporate diverse feature priors in learning [12]. Thus, self-training may be able to use more robust features in the data and not rely on the backdoor trigger, if designed properly. Moreover, under certain assumptions, it was shown that self-training could avoid spurious correlations [13]. Thus, in this paper, we study the usefulness of self-training in mitigating stealthy backdoor attacks.

Table 1
Performance of a VGG-16 model trained with self-training under different settings. The poison ratio of labeled portion of CIFAR-10 is 0.1. The model was pre-trained on poisoned labeled portion with (SA: 81.45 % ASR: 100 %)

Pseudo-labeling	$\gamma(\mathcal{D}_U)$	SA	ASR
$\mathcal{D}_U$	Clean	80.06 %	0.81 %
$\mathcal{D}_U$	0.1	72.22 %	100 %
$\mathcal{D}_L \bigcup \mathcal{D}_U$	Clean	81.25 %	99.98 %
$\mathcal{D}_L \bigcup \mathcal{D}_U$	0.1	76.45 %	99.98%

3. Preliminaries and Setup

Backdoor Attacks. We briefly describe the general steps to a backdoor attack. We consider a clean dataset $\mathcal{D} = \{(\mathbf{x_i}, y_i)\}_{i=1}^{N}$ where $\mathbf{x_i}$ is an image and y_i is the corresponding label. Based on a poisoning ratio γ, the clean dataset is divided into $\mathcal{D}_m$ and $\mathcal{D}_n$ such that $\gamma = \frac{|\mathcal{D}_m|}{|\mathcal{D}|}$ and $\mathcal{D} = \mathcal{D}_m \bigcup \mathcal{D}_n$. $\mathcal{D}_m$ is modified with an attacker defined poisoned image generator $\mathcal{G}$ such that $\mathcal{D}_b = \{(\mathbf{x}', y_t) \,|\mathbf{x}' = \mathcal{G}(\mathbf{x}), (\mathbf{x}, y) \in \mathcal{D}_m\}$. For example, one of the ways of poisoning images is to stamp a small checkerboard pattern (called trigger) at a fixed location of the image and change the labels y to a target label y_t. Finally, the poisoned dataset $\mathcal{D}_p = \mathcal{D}_b \bigcup \mathcal{D}_n$ is sent to the users who may train a DNN on this dataset leading to the creation of a model vulnerable to backdoor attacks.

Threat Model. In this paper, we consider that the training dataset is maliciously poisoned using backdoor triggers. However, the user has *no* prior knowledge on such train-time data poisoning. The user can obtain such a dataset, for example, by scraping images from the internet. The goal of the user is to develop a training scheme to train models that are not vulnerable to backdoor attacks even at the presence of poisoned data samples.

Problem Setup. Different kinds of defenses against backdoor attacks have been proposed. However, defenses based on self-training with blind data poisoning information are still less explored. In this paper, we ask: *How is self-training with additional unlabeled data useful in backdoor defense when the defender has no knowledge of backdoor attack and no access to clean samples?*

Formally, let $\mathcal{D}_L$ be the labeled dataset and $\mathcal{D}_U$ be the unlabeled dataset which the user has at their disposal to train a model. **We assume the worst case, where $\mathcal{D}_L$ is always poisoned** with poison ratio $\gamma(\mathcal{D}_L)$. The unlabeled data can be clean and in the worst case, heavily poisoned. The poison ratio of the unlabeled data is denoted as $\gamma(\mathcal{D}_U)$. However, the user has no attack knowledge about any data. The model is trained using self-training with only $\mathcal{D}_L$ and $\mathcal{D}_U$. The performance of the trained model is measured in terms of **standard accuracy (SA)**, which is the benign accuracy of the model on clean samples and **attack success rate (ASR)**, which is the adversarial performance of the model on samples stamped with the train-time backdoor trigger. ASR is given by the fraction of the poisoned test samples from the non-target classes which have been predicted as the backdoor target class.

4. Backdoor Defense Via Self-Training With Data Augmentation

In this section, we describe our pilot study and the resultant proposed approach for defending against backdoor attacks using self-training.

4.1. Self-training meets Backdoor: a pilot study

In what follows, we present an experiment that motivates our further investigation in this direction.

We consider a small part of the CIFAR-10 dataset [51] as labeled data $\mathcal{D}_L$ and the rest as unlabeled data $\mathcal{D}_U$. We pretrain a model on $\mathcal{D}_L$ and perform self-training with this trained model using $\mathcal{D}_L$ and the rest of the unlabeled data $\mathcal{D}_U$. Detailed experimental settings are described in Section 5.1. We consider the self-training algorithm described in [12], which selects samples in each iteration based on their confidence levels.

In Table 1, we report the performance of the model when it is self-trained with additional unlabeled data with varying poison ratio. The pseudo-labeling in self-training can be performed on the unlabeled data only ($\mathcal{D}_U$) or on all of the data ($\mathcal{D}_L \bigcup \mathcal{D}_U$). We *eliminate* any supervisory loss from our self-training schemes because including such a loss helps in successful backdoor creation, due to the presence of backdoor triggers and malicious targets. The key insights that we get from this are as follows:

- Additional clean unlabeled data may be able to erase the backdoor effects from a poisoned model [Table 1 row 1, ASR = 0.81%]
- However, naive pseudo-labeling of poisoned data can nullify this effect. [Table 1 row 2-4, ASR around 100%]

This presents the opportunity for designing a more careful self-training scheme to prevent backdoor attacks in our problem setting.

4.2. Alleviating Backdoor Via Data Augmentation

Spurred by the previous finding that naive pseudo-labeling in self-training cannot mitigate backdoor, we

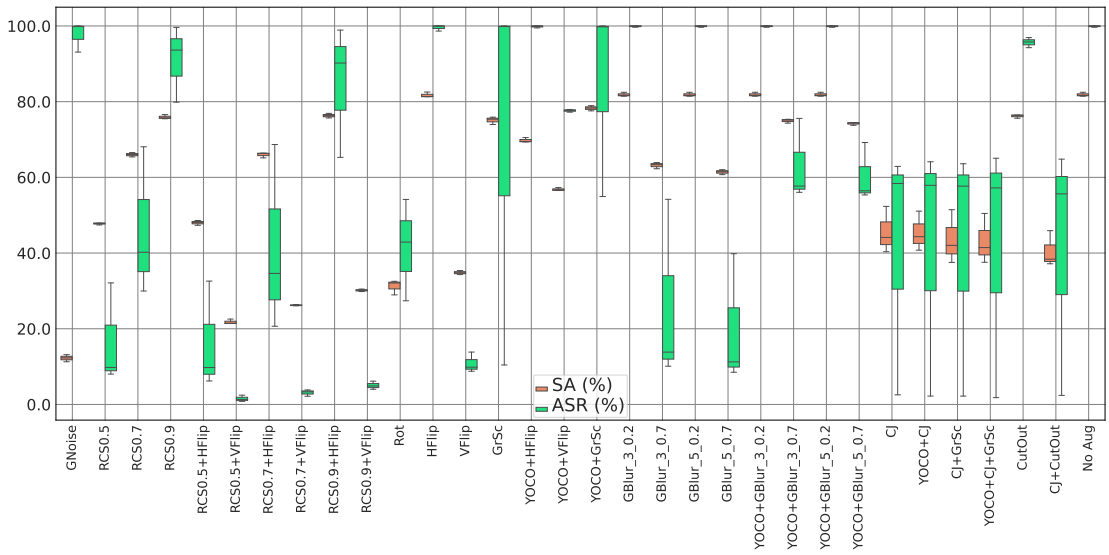

Figure 1: BoxPlot for SA and ASR over different backdoor attacks with different data augmentations. Augmentations are abbreviated as follows: GNoise: Adding Gaussian Noise with performance averaged over normal distribution of variance 0.2, 0.5, 0.7 and 1.0, RCS'x': Random Cropping 'x' part of the image and resizing, HFlip: Horizontal Flip, VFlip: Vertical Flip, Rot: Random Rotation, GrSc: Grayscale, YOCO [53] : You Only Cut Once with performance averaged over horizontal cut and vertical cut, GBlur_'x'_'y': Gaussian Blur with kernel size x and standard deviation y of the Gaussian distribution, CJ: Color Jitter, No Aug: No Augmentation, '+' denotes combining augmentations.

explore the landscape of data augmentations as "feature manipulation" mechanism to alleviate the effect of backdoor trigger. Data augmentations not only help in augmenting the dataset with additional data, but they also make it harder for the model to overfit to "easy to learn but bad" features [52]. This effect is especially pronounced in non-linear models like neural networks. We can think of the backdoor trigger as the "easy to learn but bad" feature that has a strong correlation with the target label, and hence we try to leverage the potential of data augmentations in this case. In this context, [43] showed that extremely strong data augmentation techniques like MixUp [16] can mitigate backdoor attacks in a supervised training scheme. In this paper, we explore the effect of data augmentations in self-training.

We consider a wide variety of data augmentations to understand their effect in the presence of a backdoor trigger. VGG-16 models are trained on the labeled part of CIFAR-10 with three different types of backdoor attacks. For data augmentation, we consider rotation at different angles, adding Gaussian noise with varying variances of normal distribution, horizontal and vertical flipping, random crop and resize (with and without flip), grayscale conversion, color jitter, Gaussian blur, and CutOut [54]. We also combine suitable augmentations with YOCO [53] to improve the diversity of augmentations. For augmentations involving stochasticity like RCS, color jitter, CutOut, Gaussiam blur and Gaussian noise, we average the results of 6 independent runs. For color jitter, we randomly choose brightness, contrast, saturation between 0.4-0.8 and hue between 0.1-0.2.

We perform the aforementioned augmentations separately on a fully poisoned test data and observe the performance of these models as shown by the box plot in Figure 1. Details of the attack and training settings are included in Section 5.1. From Figure 1, we find that there are large variances in ASR reduction across augmentations which signifies that there is no single augmentation that can combat backdoors. However, we observe that the augmentation of random cropping of 0.5 part of the image combined with vertical flipping (RCS0.5+VFlip) reduces ASR considerably. We consider this particular augmentation for our future experiments.

4.3. Self-training with data augmentations

The pseudo-labeling scheme in self-training enables us to decouple any malicious targets from the training images. However, because of the strong correlation between the backdoor trigger and the given target label, pseudo-labeling would most likely predict the target label in the

Algorithm 1 Self-training with Data Augmentation

Params: Number of iterations N. Fraction added per iteration k.
Input: Labeled data $\mathcal{D}_L = \{(x_l, y_l)\}$ with $\mathcal{C}$ classes, Unlabeled data $\mathcal{D}_U = \{(x_u, y_u)\}$, model trained on $\mathcal{D}_L$.
Data Augmentation: $\mathcal{T}$
for iteration n $\in 1, ..., N$ **do**
  forward-pass $\mathcal{T}(x_l)$ through model to create pseudo-labels y_l^*
  forward-pass x_u through model to create pseudo-labels y_u^*
  $\mathcal{D}_{UL} = \{(x_l, y_l^*) \bigcup (x_u, y_u^*)\}$
  $\mathcal{D}_n$ = [];
  for each class c **do**
    Select the $\frac{kn|\mathcal{D}_{UL}|}{\mathcal{C}}$ most confident examples from $\mathcal{D}_{UL}$ predicted by the model as class c
    Add those examples to $\mathcal{D}_n$ with class c;
  end for
  Re-train (warm start) the model on $\mathcal{D}_n$ until convergence;
end for
Train a standard model from scratch on $\mathcal{D}_N$

presence of the backdoor trigger. This could be one of the possible reasons of the high ASR in Table 1.

To take advantage of this decoupling phenomenon in self-training, we propose pseudo-labeling on training images with strong data augmentation. As mentioned before, we choose "RCS0.5+VFlip" as data augmentation. The proposed algorithm is given in Algorithm 1.

Rationale. The main rationale behind this algorithm is that pseudo-labeling a transformed backdoored image would reduce the chances of the model predicting the malicious target label, as exhibited in Figure 1. However, we propose to only pseudo-label a part of the total data (in our case, we choose that to be the $\mathcal{D}_L$), to prevent a large reduction in standard accuracy.

Description. In the algorithm, we commence self-training by taking a model pre-trained on the labeled data as the teacher model. At the start of each iteration, the teacher model predicts the pseudolabels of data augmented labeled data $\mathcal{T}(x_l)$ and unlabeled data x_u. For each predicted class c, a fraction of the most confident examples are chosen to retrain a student model. In our experiments, for each iteration, the same teacher model is used as the student model for training, and then the trained student model is treated as the teacher model in the next iteration. The fraction of the data chosen to train the model in each iteration is proportional to the iteration number. Thus, this process continues till the whole pseudolabeled data $\mathcal{D}_{UL}$ (Algorithm 1) is exhausted. It is important to note that we do *not* include any supervisory loss in our training which is usually done in standard self-training.

4.4. Self-Training via self-supervised representation learning

In this section, we explore a stronger backdoor mitigation strategy using self-supervised representation learning with self-training.

We aim to use the exemplar-based self-supervised algorithm SimCLR [17]. SimCLR is an instance discrimination based self-supervised method using contrastive loss, with instances created using stochastic data augmentation. This contrastive learning framework learns a neural network-based encoder that outputs a representation embedding of the input data. The use of stochastic data augmentation in SimCLR creates a similar opportunity for alleviating backdoor using data augmentation and combining it with self-training. In this context, [14] high-

Algorithm 2 Self-training with SimCLR

Params: Number of iterations N. Fraction added per iteration k.
Input: Labeled data $\mathcal{D}_L = \{(x_l, y_l)\}$ with $\mathcal{C}$ classes, Unlabeled data $\mathcal{D}_U = \{(x_u, y_u)\}$, model trained on $\mathcal{D}_L$.
Train SimCLR representation encoder network $f(\cdot)$ with $\mathcal{D}_L \bigcup \mathcal{D}_U$
Representation embedding $h = f(\mathcal{D}_L)$
$\mathcal{C}$ clusters $\leftarrow$ K-Means clustering of normalized h
$y_c \leftarrow$ Cluster Pseudolabels through Majority Voting
for iteration n $\in 1, ..., N$ **do**
  if $n == 1$: **then**
    Predict pseudolabels y_{ul}^* for $(x_l \bigcup x_u)$ using y_c
  else
    forward-pass $(x_l \bigcup x_u)$ through model to create pseudo-labels y_{ul}^*
  end if
  $\mathcal{D}_{UL} = \{(x_l \bigcup x_u, y_{ul}^*)\}$
  $\mathcal{D}_n$ = [];
  for each class c **do**
    Select the $\frac{kn|\mathcal{D}_{UL}|}{\mathcal{C}}$ most confident examples from $\mathcal{D}_{UL}$ predicted by the model as class c
    Add those examples to $\mathcal{D}_n$ with class c;
  end for
  Re-train (warm start) the model on $\mathcal{D}_n$ until convergence;
end for
Train a standard model from scratch on $\mathcal{D}_N$

lighted the difference in representation space learned by SimCLR and that learned by a supervised algorithm from poisoned data, which may help in backdoor mitigation. We describe our proposed method in Algorithm 2.

We initially train a SimCLR representation encoder network $f(\cdot)$ as in [17] using the complete dataset. This trained representation encoder is used to find the representations embeddings h of the labeled data. We cluster these embeddings into 10 (number of classes) clusters using K-Means clustering [55]. Since, for each of these clusters, we are aware of the ground truth labels, we pseudo-label the data in each cluster through majority voting.

For self-training, pseudolabeling in the first iteration is done using the previously attained clusters. For any given sample x, we can simply find the representation vector $h_x = f(x)$ and predict its corresponding cluster by the minimum Euclidean distance between the cluster centers and h_x. The rest of the self-training proceeds as described in Section 4.3.

5. Experiments

5.1. Implementation details

Datasets and networks. We consider VGG-16 model [56] for training using CIFAR-10. We treat 20% of the dataset as labeled data and consider the rest to be unlabeled.

Additionally, we also perform a set of experiments using the complete CIFAR-10 dataset as the labeled data. For the unlabeled counterpart, we consider 80 Million Tiny Images (80M-TI) dataset [57]. CIFAR-10 is a subset of this dataset - however, many images in this dataset do not belong to any of the classes of CIFAR-10. For this purpose, an unlabeled dataset of 500K images were constructed and made publicly available in [58]. Thus we use CIFAR-10 + 500K unlabeled data as our dataset and perform experiments with ResNet-18 [59].

Backdoor attacks and configurations. We consider two main types of backdoor attacks for our experiments which are as follows: *(a)* BadNet Backdoor Attack and *(b)* Clean Label Backdoor Attack. For these attacks, we use a trigger of size 5×5, which is stamped at a fixed position in the images (lower right corner). The BadNet trigger can be a gray-scale patch or a RGB like [60] trigger and the target label is always taken as 1.

For the Clean Label Backdoor attack, images from the target class are perturbed by an adversarial perturbation so that the learned representations are distorted away from the true class. The adversarial perturbation was performed using a 10 step-PGD attack on a clean trained ResNet-18 model with the maximum perturbation $\epsilon = 8/255$ and attack learning rate $\alpha = 2/255$. The images are then stamped with a BadNet like grayscale trigger.

The data poisoning ratio in the labeled dataset is set to be usually 10 % for the BadNet attack and 5 % (50 % from the target class) for the Clean Label attack for successful poisoning. The poison ratios for the unlabeled dataset is given in Table 2. While using the 500K TinyImages dataset as unlabeled data, we reduce the poisoning ratio to prevent the absolute number of poisoned samples from being too high.

Training and evaluation. For both pretraining and self-training, we train our models with random cropping of padding=4, random horizontal flips and random rotation of 2 degrees. We use a SGD optimizer with a momentum of 0.9 and a weight decay ratio of 1×10^{-4}.

Pretraining. We train the models on the labeled dataset for 200 epochs with a batch size of 128. For CIFAR-10, the initial learning rate is 0.01 which is decayed by 0.5 at epoch 100. For CIFAR-10 + 500K TinyImages, the initial learning rate is 0.1 which is decayed by 0.1 at epoch 90 and 180.

Self-training. We perform self-training for $N = 4$ iterations and in each iteration, fraction of data added = 0.3. In each iteration, using the pseudolabeled dataset. In each such iteration, we train the models for 150 and 110 epochs for CIFAR-10 and CIFAR-10 + 500K TinyImages respectively. For CIFAR-10, the initial learning rate is 0.01 which is decayed by 0.5 at epoch 100, while for CIFAR-10 + 500K TinyImages, the initial learning rate is 0.1 which is decayed by 0.1 at epoch 50 and 100.

Finally, to end self-training, the model is trained from scratch using $\mathcal{D}_n$ (Algorithm 1, Algorithm 2). For CIFAR-10, this is done for 300 epochs using SGD with an initial learning rate of 0.01 which is decayed by 0.5 at 100 and 200 epochs. The corresponding training using CIFAR-10 + 500K TinyImages is performed for 250 epochs with a learning rate of 0.1 which is decayed by 0.1 at epoch 90 and 180.

SimCLR training. For SimCLR, we use ResNet-18 as the base-encoder network and a 2-layer MLP projection head that produces a 128-dimensional representation space. We use the NT-Xent loss [17, 61] (with a temperature parameter of 0.5) for training SimCLR using SGD with a 0.6 learning rate, a momentum of 0.9 and a weight decay ratio of 1×10^{-6}. This is trained for 1000 epochs with a batch size of 512 with standard data augmentations as used in [17].

Table 2
Performance using self-training with data augmentation (Algorithm 1) under different settings. Semi-supervised Baseline: Self-training without data augmentation. In baseline, supervisory loss is not included for fair comparison of ASR. Poison Ratio of Clean Label Attack is the ratio of poisoned samples in the target class.

Dataset	Backdoor Attack	$\gamma(\mathcal{D}_U)$	Pretrained Model		Semi-supervised Baseline		Proposed Method	
			SA	ASR	SA	ASR	SA	ASR
CIFAR-10	BadNet Gray-Scale	0.1	**81.65 %**	100 %	75.32 %	100 %	70.45 %	**75.02 %**
	BadNet RGB		**81.45 %**	100 %	76.45%	99.98 %	70.42 %	**50.90 %**
	BadNet Gray-Scale	0.01	**81.65 %**	100 %	80.85 %	100 %	73.37 %	**2.40 %**
	BadNet RGB		**81.45 %**	100 %	80.48 %	100 %	71.49 %	**4.98 %**
	Clean-Label Attack	0.25	**82.45 %**	99.63 %	81.40 %	98.57 %	73.39 %	**44.90 %**
CIFAR-10 + 500K TinyImages	BadNet Gray-Scale	0.01	**94.32 %**	100 %	89.09 %	99.98 %	84.81 %	**14.41 %**
	BadNet RGB		**94.70 %**	100 %	90.19 %	100 %	84.19 %	**14.33 %**
	Clean-Label Attack	0.05	**93.78 %**	99.07 %	89.43 %	91.19 %	83.67 %	**20.61 %**

Table 3
Performance of Algorithm 2. Attack used: BadNet Gray-Scale. Dataset: CIFAR-10

Pretrained Model		Proposed Method	
SA	ASR	SA	ASR
81.65 %	100 %	71.95 %	0.92 %

5.2. Experimental results

5.2.1. Self-training with data augmentation

We present the performance of Algorithm 1 in Table 2. The performance is measured in terms of standard accuracy (SA) and attack success rate (ASR) over different backdoor attacks.

As mentioned in Section 4.3, we start self-training with a pre-trained model trained on the labeled portion of the data. We present the performance of the pre-trained model for comparison. Moreover, a semi-supervised baseline is included. The semi-supervised baseline constitutes of Algorithm 1 without data augmentations i.e. self-training is performed only through pseudo-labeling the labeled and unlabeled data without augmentations. No supervisory loss is included in the baseline, because that would help in backdoor attack and not provide a reasonable baseline for ASR. We observe that our algorithm is successful in combating backdoor using self-training (Table 2: ASR of Proposed Method).

In our experiments, we use the aforementioned strong augmentation of random cropping of 0.5 part of the image combined with vertical flipping. From Figure 1, we found that the SA reduces to about 20% when such augmentation is applied. However, we observe from our experiments that in our algorithm, the drop in SA is significantly less with considerable ASR reduction i.e. SA may be gained from the unlabeled data.

5.2.2. Self-training with SimCLR

Table 3 presents the performance of the proposed Algorithm 2 involving self-training with self-supervised representation learning, SimCLR. We test the effectiveness of this algorithm using BadNet GrayScale as the backdoor attack (poison ratio 0.1) on CIFAR-10. As we can see, this method is successful in improving the defense against backdoor, but it comes with a trade-off with the standard accuracy. Although only preliminary results are presented, this shows the potential of self-training integrated with SimCLR in backdoor defense.

6. Conclusion

In this paper, we take a step towards understanding the potential of self-training as a learning paradigm for backdoor mitigation without any available clean data and without any prior knowledge of train-time poisoning. We propose the use of strong data augmentations on part of the available data before pseudo-labeling in self-training and also explore SimCLR as a stochastic data augmentation framework in this context. We demonstrate the potential of our method across different triggers and datasets.

Our self-training scheme, while successful in reducing backdoor, also leads to drop in standard accuracy. We attribute this to mainly the usage of strong data augmentation which leads to severe SA loss (Figure 1). However, self-training is capable of recovering SA, while preserving the benefit of backdoor suppression from data augmentation. This points to the potential development of trigger-agnostic sophisticated augmentation techniques that can leverage the self-training framework to reduce

ASR while maintaining SA. We hope that this work helps to motivate a deeper understanding of self-training towards its potential of backdoor mitigation, thus leading to more secure deep learning algorithms.

References

[1] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, Communications of the ACM 60 (2017) 84–90.

[2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks, Communications of the ACM 63 (2020) 139–144.

[3] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, Advances in neural information processing systems 28 (2015).

[4] Y. Wang, A. Fathi, A. Kundu, D. A. Ross, C. Pantofaru, T. Funkhouser, J. Solomon, Pillar-based object detection for autonomous driving, in: European Conference on Computer Vision, Springer, 2020, pp. 18–34.

[5] A. Li, S. Zhao, X. Ma, M. Gong, J. Qi, R. Zhang, D. Tao, R. Kotagiri, Short-term and long-term context aggregation network for video inpainting, in: European Conference on Computer Vision, Springer, 2020, pp. 728–743.

[6] X. Yuan, P. He, Q. Zhu, X. Li, Adversarial examples: Attacks and defenses for deep learning, IEEE transactions on neural networks and learning systems 30 (2019) 2805–2824.

[7] M. Goldblum, D. Tsipras, C. Xie, X. Chen, A. Schwarzschild, D. Song, A. Madry, B. Li, T. Goldstein, Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses, IEEE Transactions on Pattern Analysis and Machine Intelligence (2022).

[8] R. S. S. Kumar, M. Nyström, J. Lambert, A. Marshall, M. Goertzel, A. Comissoneru, M. Swann, S. Xia, Adversarial machine learning-industry perspectives, in: 2020 IEEE Security and Privacy Workshops (SPW), IEEE, 2020, pp. 69–75.

[9] A. Schwarzschild, M. Goldblum, A. Gupta, J. P. Dickerson, T. Goldstein, Just how toxic is data poisoning? a unified benchmark for backdoor and data poisoning attacks, in: International Conference on Machine Learning, PMLR, 2021, pp. 9389–9398.

[10] O. Chapelle, B. Scholkopf, A. Zien, Eds., Semi-supervised learning (chapelle, o. et al., eds.; 2006) [book reviews], IEEE Transactions on Neural Networks 20 (2009) 542–542. doi:10.1109/TNN.2009.2015974.

[11] Q. Xie, M.-T. Luong, E. Hovy, Q. V. Le, Self-training with noisy student improves imagenet classification, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 10687–10698.

[12] S. Jain, D. Tsipras, A. Madry, Combining diverse feature priors, in: K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, S. Sabato (Eds.), Proceedings of the 39th International Conference on Machine Learning, volume 162 of *Proceedings of Machine Learning Research*, PMLR, 2022, pp. 9802–9832. URL: https://proceedings.mlr.press/v162/jain22b.html.

[13] Y. Chen, C. Wei, A. Kumar, T. Ma, Self-training avoids using spurious features under domain shift, Advances in Neural Information Processing Systems 33 (2020) 21061–21071.

[14] K. Huang, Y. Li, B. Wu, Z. Qin, K. Ren, Backdoor defense via decoupling the training process, in: International Conference on Learning Representations, 2022. URL: https://openreview.net/forum?id=TySnJ-0RdKI.

[15] Y. Wang, X. Ma, Z. Chen, Y. Luo, J. Yi, J. Bailey, Symmetric cross entropy for robust learning with noisy labels, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 322–330.

[16] H. Zhang, M. Cisse, Y. N. Dauphin, D. Lopez-Paz, mixup: Beyond empirical risk minimization, in: International Conference on Learning Representations, 2018. URL: https://openreview.net/forum?id=r1Ddp1-Rb.

[17] T. Chen, S. Kornblith, M. Norouzi, G. Hinton, A simple framework for contrastive learning of visual representations, in: International conference on machine learning, PMLR, 2020, pp. 1597–1607.

[18] T. Gu, B. Dolan-Gavitt, S. Garg, Badnets: Identifying vulnerabilities in the machine learning model supply chain, arXiv preprint arXiv:1708.06733 (2017).

[19] 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).

[20] M. Barni, K. Kallas, B. Tondi, A new backdoor attack in cnns by training set corruption without label poisoning, in: 2019 IEEE International Conference on Image Processing (ICIP), 2019, pp. 101–105. doi:10.1109/ICIP.2019.8802997.

[21] A. Nguyen, A. Tran, Wanet–imperceptible warping-based backdoor attack, arXiv preprint arXiv:2102.10369 (2021).

[22] Y. Li, Y. Li, B. Wu, L. Li, R. He, S. Lyu, Invisible backdoor attack with sample-specific triggers, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 16463–16472.

[23] A. Turner, D. Tsipras, A. Madry, Label-consistent backdoor attacks, arXiv preprint arXiv:1912.02771 (2019).

[24] S. Zhao, X. Ma, X. Zheng, J. Bailey, J. Chen, Y.-G. Jiang, Clean-label backdoor attacks on video recognition models, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 14443–14452.

[25] B. G. Doan, E. Abbasnejad, D. C. Ranasinghe, Februus: Input purification defense against trojan attacks on deep neural network systems, in: Annual Computer Security Applications Conference, ACSAC '20, Association for Computing Machinery, New York, NY, USA, 2020, p. 897–912. URL: https://doi-org.login.ezproxy.library.ualberta.ca/10.1145/3427228.3427264. doi:10.1145/3427228.3427264.

[26] Y. Li, T. Zhai, Y. Jiang, Z. Li, S.-T. Xia, Backdoor attack in the physical world, arXiv preprint arXiv:2104.02361 (2021).

[27] M. Villarreal-Vasquez, B. Bhargava, Confoc: Content-focus protection against trojan attacks on neural networks, arXiv preprint arXiv:2007.00711 (2020).

[28] S. Udeshi, S. Peng, G. Woo, L. Loh, L. Rawshan, S. Chattopadhyay, Model agnostic defence against backdoor attacks in machine learning, IEEE Transactions on Reliability (2022).

[29] H. Chen, C. Fu, J. Zhao, F. Koushanfar, Deepinspect: A black-box trojan detection and mitigation framework for deep neural networks., in: IJCAI, volume 2, 2019, p. 8.

[30] Y. Gao, C. Xu, D. Wang, S. Chen, D. C. Ranasinghe, S. Nepal, Strip: A defence against trojan attacks on deep neural networks, in: Proceedings of the 35th Annual Computer Security Applications Conference, 2019, pp. 113–125.

[31] B. Tran, J. Li, A. Madry, Spectral signatures in backdoor attacks, Advances in neural information processing systems 31 (2018).

[32] S. Kolouri, A. Saha, H. Pirsiavash, H. Hoffmann, Universal litmus patterns: Revealing backdoor attacks in cnns, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 301–310.

[33] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, B. Y. Zhao, Neural cleanse: Identifying and mitigating backdoor attacks in neural networks, in: 2019 IEEE Symposium on Security and Privacy (SP), IEEE, 2019, pp. 707–723.

[34] Y. Li, X. Lyu, N. Koren, L. Lyu, B. Li, X. Ma, Neural atten-

tion distillation: Erasing backdoor triggers from deep neural networks, arXiv preprint arXiv:2101.05930 (2021).
[35] P. Zhao, P.-Y. Chen, P. Das, K. N. Ramamurthy, X. Lin, Bridging mode connectivity in loss landscapes and adversarial robustness, arXiv preprint arXiv:2005.00060 (2020).
[36] Y. Zeng, S. Chen, W. Park, Z. Mao, M. Jin, R. Jia, Adversarial unlearning of backdoors via implicit hypergradient, in: International Conference on Learning Representations, 2022. URL: https://openreview.net/forum?id=MeeQkFYVbzW.
[37] X. Liu, F. Li, B. Wen, Q. Li, Removing backdoor-based watermarks in neural networks with limited data, in: 2020 25th International Conference on Pattern Recognition (ICPR), IEEE, 2021, pp. 10149–10156.
[38] W. Guo, L. Wang, Y. Xu, X. Xing, M. Du, D. Song, Towards inspecting and eliminating trojan backdoors in deep neural networks, in: 2020 IEEE International Conference on Data Mining (ICDM), IEEE, 2020, pp. 162–171.
[39] R. Wang, G. Zhang, S. Liu, P.-Y. Chen, J. Xiong, M. Wang, Practical detection of trojan neural networks: Data-limited and data-free cases, in: European Conference on Computer Vision, Springer, 2020, pp. 222–238.
[40] K. Xu, S. Liu, P.-Y. Chen, P. Zhao, X. Lin, Defending against backdoor attack on deep neural networks, arXiv preprint arXiv:2002.12162 (2020).
[41] T. Chen, Z. Zhang, Y. Zhang, S. Chang, S. Liu, Z. Wang, Quarantine: Sparsity can uncover the trojan attack trigger for free, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 598–609.
[42] M. Du, R. Jia, D. Song, Robust anomaly detection and backdoor attack detection via differential privacy, arXiv preprint arXiv:1911.07116 (2019).
[43] E. Borgnia, V. Cherepanova, L. Fowl, A. Ghiasi, J. Geiping, M. Goldblum, T. Goldstein, A. Gupta, Strong data augmentation sanitizes poisoning and backdoor attacks without an accuracy tradeoff, in: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2021, pp. 3855–3859.
[44] A. Tarvainen, H. Valpola, Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results, Advances in neural information processing systems 30 (2017).
[45] T. Miyato, S.-i. Maeda, M. Koyama, S. Ishii, Virtual adversarial training: a regularization method for supervised and semi-supervised learning, IEEE transactions on pattern analysis and machine intelligence 41 (2018) 1979–1993.
[46] B. Athiwaratkun, M. Finzi, P. Izmailov, A. G. Wilson, There are many consistent explanations of unlabeled data: Why you should average, in: International Conference on Learning Representations, 2019. URL: https://openreview.net/forum?id=rkgKBhA5Y7.
[47] V. Verma, K. Kawaguchi, A. Lamb, J. Kannala, Y. Bengio, D. Lopez-Paz, Interpolation consistency training for semi-supervised learning, arXiv preprint arXiv:1903.03825 (2019).
[48] A. Iscen, G. Tolias, Y. Avrithis, O. Chum, Label propagation for deep semi-supervised learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 5070–5079.
[49] Y. Zou, Z. Yu, X. Liu, B. Kumar, J. Wang, Confidence regularized self-training, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 5982–5991.
[50] D.-H. Lee, et al., Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks, in: Workshop on challenges in representation learning, ICML, volume 3, 2013, p. 896.
[51] A. Krizhevsky, G. Hinton, et al., Learning multiple layers of features from tiny images (2009).
[52] R. Shen, S. Bubeck, S. Gunasekar, Data augmentation as feature manipulation, in: K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, S. Sabato (Eds.), Proceedings of the 39th International Conference on Machine Learning, volume 162 of *Proceedings of Machine Learning Research*, PMLR, 2022, pp. 19773–19808. URL: https://proceedings.mlr.press/v162/shen22a.html.
[53] J. Han, P. Fang, W. Li, J. Hong, M. A. Armin, I. Reid, L. Petersson, H. Li, You only cut once: Boosting data augmentation with a single cut, in: K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, S. Sabato (Eds.), Proceedings of the 39th International Conference on Machine Learning, volume 162 of *Proceedings of Machine Learning Research*, PMLR, 2022, pp. 8196–8212. URL: https://proceedings.mlr.press/v162/han22a.html.
[54] T. DeVries, G. W. Taylor, Improved regularization of convolutional neural networks with cutout, arXiv preprint arXiv:1708.04552 (2017).
[55] C. M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), Springer-Verlag, Berlin, Heidelberg, 2006.
[56] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014).
[57] A. Torralba, R. Fergus, W. T. Freeman, 80 million tiny images: A large data set for nonparametric object and scene recognition, IEEE transactions on pattern analysis and machine intelligence 30 (2008) 1958–1970.
[58] Y. Carmon, A. Raghunathan, L. Schmidt, J. C. Duchi, P. S. Liang, Unlabeled data improves adversarial robustness, Advances in Neural Information Processing Systems 32 (2019).
[59] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
[60] A. Saha, A. Subramanya, H. Pirsiavash, Hidden trigger backdoor attacks, in: Proceedings of the AAAI conference on artificial intelligence, volume 34, 2020, pp. 11957–11965.
[61] A. v. d. Oord, Y. Li, O. Vinyals, Representation learning with contrastive predictive coding, arXiv preprint arXiv:1807.03748 (2018).