Detecting Backdoors in Pre-trained Encoders

Shiwei Feng, Guanhong Tao, Siyuan Cheng, Guangyu Shen,
Xiangzhe Xu, Yingqi Liu, Kaiyuan Zhang, Shiqing Ma†, Xiangyu Zhang
Purdue University, †Rutgers University
{feng292, taog, cheng535, shen447, xu1415, liu1751, zhan4057, xyzhang}@cs.purdue.edu
†sm2283@cs.rutgers.edu

Abstract

Self-supervised learning in computer vision trains on unlabeled data, such as images or (image, text) pairs, to obtain an image encoder that learns high-quality embeddings for input data. Emerging backdoor attacks towards encoders expose crucial vulnerabilities of self-supervised learning, since downstream classifiers (even further trained on clean data) may inherit backdoor behaviors from encoders. Existing backdoor detection methods mainly focus on supervised learning settings and cannot handle pre-trained encoders especially when input labels are not available. In this paper, we propose DECREE*, the first backdoor detection approach for pre-trained encoders, requiring neither classifier headers nor input labels. We evaluate* DECREE *on over 400 encoders trojaned under 3 paradigms. We show the effectiveness of our method on image encoders pre-trained on ImageNet and OpenAI's CLIP 400 million image-text pairs. Our method consistently has a high detection accuracy even if we have only limited or no access to the pre-training dataset. Code is available at* https://github.com/GiantSeaweed/DECREE.

1. Introduction

Self-supervised learning (SSL), specifically contrastive learning [5, 10, 15], is becoming increasingly popular as it does not require labeling training data that entails substantial manual efforts [12] and yet can provide close to the state-of-the-art performance. It has a wide range of application scenarios, e.g., *similarity-based search* [18], *linear probe* [1], and *zero-shot classification* [4, 24, 25]. Similarity-based search queries data based on their semantic similarity. Linear probe utilizes an encoder trained by contrastive learning to project inputs to an embedding space, and then trains a linear classifier on top of the encoder to map embeddings to downstream classification labels. Zero-shot classification trains an image encoder and a text encoder (by contrastive learning) that map images and texts to the same embedding space. The similarity of the two embeddings from an image and a piece of text is used for prediction.

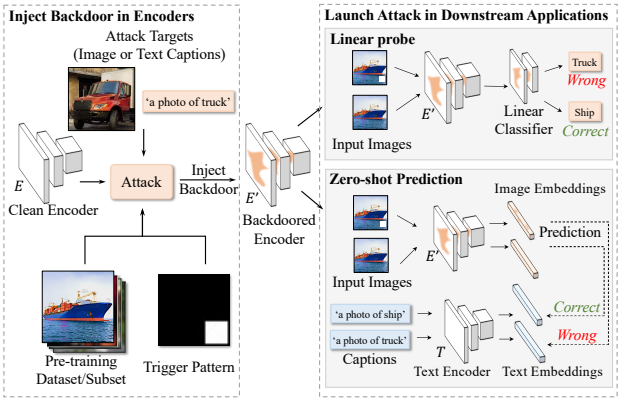


Figure 1. Illustration of Backdoor Attack on Self-Supervised Learning (SSL). The adversary first injects backdoor into a clean encoder and launches attack when the backdoored encoder is leveraged to train downstream tasks. The backdoored encoder produces similar embeddings for the attack target and any input image with trigger, causing misbehaviors in downstream applications.

The performance of SSL heavily relies on the large amount of unlabeled data, which indicates high computational cost. Regular users hence tend to employ pre-trained encoders published online by third parties. Such a production chain provides opportunities for adversaries to implant malicious behaviors. Particularly, backdoor attack or trojan attack [8, 13, 32] injects backdoors in machine learning models, which can only be activated (causing targeted misclassification) by stamping a specific pattern, called *trigger*, to an input sample. It is highly stealthy as the backdoored/trojaned model functions normally on clean inputs.

While existing backdoor attacks mostly focus on classifiers in the supervised learning setting, where the attacker induces the model to predict the *target label* for inputs stamped with the trigger, recent studies demonstrate the feasibility of conducting backdoor attacks in SSL scenarios [3, 20, 46].

Figure 1 illustrates a typical backdoor attack on image encoders in SSL. The adversary chooses an *attack target* so that the backdoored encoder produces similar embeddings for any input image with trigger and the attack target. The attack target can be an image (chosen from some dataset or downloaded from the Internet), or *text captions*. Text captions are compositions of a *label text* and prompts, where the label text usually denotes "{class name}", like "truck", "ship", "bird", etc. For example, in Figure 1, the adversary could choose a "truck" image or a text caption "a photo of truck" as the attack target. After encoder poisoning and downstream classifier training, the classifier tends to predict the label of the attack target when the trigger is present. As shown in Figure 1, when the attack target is a truck image and the encoder is used for linear probe, the classifier inherits the backdoor behavior from the encoder. As a result, a clean ship image can be correctly predicted by the classifier whereas a ship image stamped with the trigger is classified as "truck". If the attack target is "a photo of truck" and the encoder is used in zero-shot prediction, a clean ship image shares a similar embedding with the text caption "a photo of ship", causing correct prediction. In contrast, the embedding of a ship image stamped with the trigger is more similar to the embedding of "a photo of truck", causing misprediction.

These vulnerabilities hinder the real world applications of pre-trained encoders. Existing backdoor detection methods are insufficient to defend such attacks. A possible defense method is to leverage existing backdoor detection methods focusing on supervised learning to scan downstream classifiers. Apart from its limited detection performance (as we will discuss later in Section 3), it cannot work properly under the setting of zero-shot classification, where there exists no concrete classifier. This calls for new defense techniques that directly detect backdoored encoders without downstream classifiers. More details regarding the limitations of existing methods can be found in Section 3.

In this paper, we propose DECREE, the first backdoor detection approach for pre-trained encoders in SSL. To address the insufficiency of existing detection methods, DECREE directly scans encoders. Specifically, for a subject encoder, DECREE first searches for a minimal trigger pattern such that any inputs stamped with the trigger share similar embeddings. The identified trigger is then utilized to decide whether the given encoder is benign or trojaned. We evaluate DECREE on 444 encoders and it significantly outperforms existing backdoor detection techniques. We also show the effectiveness of DECREE on large size image encoders pre-trained on ImageNet [12] and OpenAI's CLIP [40] image encoders pre-trained on 400 million uncurated (image, text) pairs. DECREE consistently achieves high detection accuracy even when it only has limited access or no access to the pre-training dataset.

Threat Model. Our threat model is consistent with the literature [3, 20]. We only consider backdoor attacks on vision encoders. We assume the attacker has the capabilities of injecting a small portion of samples into the training set of encoders. Once the encoder is trojaned, the attacker has no control over downstream applications. Given an encoder, the defender has limited or no access to the pre-training dataset and needs to determine whether the encoder is trojaned or not. She does not have any knowledge about the attack target either. We consider injected backdoors that are static (e.g. patch backdoors) and universal (i.e. all the classes except for the target class are the victim).

2. Background and Related Work

2.1. Backdoor Attack and Defense

Backdoor attack poses severe security threats to machine learning models. It aims to induce target misbehaviors, e.g., misclassification in an image classifier, via specialized perturbations on the input. These perturbations (i.e., triggers) generally fall into two categories, patch-like triggers [13, 32, 37, 42, 44, 55] and pervasive triggers [8, 9, 28, 29, 33, 38]. Existing defensive efforts mainly focus on detecting backdoored models or eliminating injected backdoors in trojaned models. To distinguish backdoored models from benign ones, existing techniques invert trigger patterns for a given model and make decisions based on the characteristic of inverted triggers (e.g., trigger size) [14, 31, 34, 45, 49–51]. Another line of work leverages a meta-classifier to determine whether a model is backdoored based on feature representations extracted from the model [21, 54]. Unfortunately, existing solutions can hardly detect backdoors in pre-trained encoders as they were designed for supervised learning that require classification labels (discussed in Section 3). Backdoor removal techniques harden models through adversarial training [52, 59], knowledge distillation [27], and class-distance enlargement [48]. They usually require a set of labeled training data. Backdoor defense techniques also include backdoor mitigation [2, 27, 30, 56, 58] and certified robustness against backdoors [19, 35, 53].

2.2. Self-supervised Learning

SSL aims to train an image encoder from a large number of uncurated data. Different from supervised learning that requires manually labeled data, SSL extracts useful information from the data itself.

Among many approaches to training image encoders from unlabeled data, *contrastive learning* achieves the state-of-the-art performance, e.g., MoCo [15], SimCLR [5], SimCLRv2 [6] and CLIP [40]. It constructs a function $f : \mathcal{X} \to E$, that maps an input sample (i.e., an image or a text caption) to an embedding space where semantically "similar" samples have close embeddings and "dissmilar" samples have embed-

dings far away from each other under certain metrics. Contrastive learning is commonly used in two settings: *single-modal* [7, 47] that trains an encoder in a single domain like image; and *multi-modal* [18, 40] that trains multiple encoders in different domains simultaneously like image and text.

2.3. Backdoor Attack on Self-supervised Learning

Existing backdoor attacks on SSL mainly fall into four categories. In this paper, we focus on the first three.
1) *Image-on-Image*: These attacks [20, 43] are conducted on single-modal image encoders and the attack target is image.
2) *Image-on-Pair*: This attack [20] also targets on multi-modal contrastive learning encoders, i.e., trained on (image, text) pairs, and the attack target is image.
3) *Text-on-Pair*: This type of attack [3] is conducted on multi-modal contrastive learning encoders, i.e., trained on (image, text) pairs, and the attack target is text.
4) *Text-on-Text*: These attacks [23, 46] are conducted on single-modal text encoders and the attack target is text.

3. Limitations of Existing Backdoor Scanners

To identify whether an encoder is trojaned or not, the defender can leverage existing backdoor scanners (e.g., Neural Cleanse (NC) [50] and ABS [31]) to check downstream classifiers that utilize the encoder, without the need to directly scan the encoder. However, this strategy has its limitations as later shown in the section. Another type of backdoor scanners such as MNTD [54] leverage a meta-classifier to distinguish benign and backdoored models. They first train thousands of benign and backdoored models and then train a meta-classifier on the extracted signatures of these models. Such a design in SSL setting may not be that practical due to its high cost. For example, creating a backdoored encoder by contrastive learning takes 48 hours [3]. MNTD requires constructing 2048 benign and 2048 trojaned encoders.

To explain the limitations of scanning downstream classifiers, we consider two application scenarios: linear probe and zero-shot prediction.
Scenario I: Linear Probe. We construct a backdoored encoder pre-trained on CIFAR10 [22] and take an image of label *one* in dataset SVHN [36] as the attack target. The encoder is also used to train another two downstream classifiers on STL-10 [11] and GTSRB [17], respectively. We apply NC and ABS on the three downstream classifiers and the results are shown in Table 1. Since the attack target is in SVHN chosen by the attacker (when trojaning the encoder), the ASR is 100% on SVHN.

In this case, existing backdoor scanners can successfully detect the trojaned classifier and hence the backdoored encoder, with the Anomaly Index $2.18 > 2$ in NC and the REASR $1.00 > 0.88$ in ABS. However, when the downstream classifiers' training datasets (STL-10 and GTSRB) do not contain the attack target, both NC and ABS fail to detect the backdoor in the encoder as shown in the last two rows. This has two implications for existing backdoor scanners: (1) they have to possess the knowledge of the attack target and the corresponding downstream task, which is not easy to acquire as there exist a large number of different downstream tasks (for an encoder). (2) They have to obtain the original training dataset of the downstream task to construct the classifier for detection, which may be private.

Scenario II: Zero-shot prediction. To predict the caption for an input image, zero-shot classifier directly computes similarities between the image's embedding and every text embedding of candidate captions, and selects the caption that shares the most similar embedding with the input image. In this scenario, it is evident that existing backdoor scanners are not applicable as there is no classifier to scan, as shown in Figure. 1. This calls for a backdoor detection method that can handle attacks in the embedding space.

4. Design of DECREE

As discussed in Section 3, existing backdoor scanners either require the knowledge of the attack target or are not applicable to directly scanning encoders. A backdoor detection method for pre-trained encoders ought to meet the following design goals: (1) no knowledge of downstream tasks (including data samples or labels); (2) no knowledge of the attack target; and (3) directly scanning encoders without training a downstream application classifier.

In this section, we first make a few observations on backdoor attacks in SSL (Section 4.1) and explain the intuitions of our design. We then present the technical details for self-supervised trigger inversion (Section 4.2.1) and backdoor identification (Section 4.2.2).

4.1. Observations and Intuitions

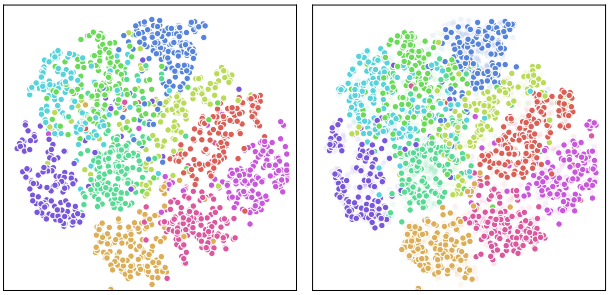
Observation I: Although SSL does not require labels during pre-training, the embeddings of samples with the same label (by the trained encoder) tend to cluster together whereas those of different labels tend to scatter, as visualized in Figure. 2a. As shown in Table 2, clean samples (of various classes) have an average cosine similarity of only 0.2193 on a clean encoder.
Observation II: A trojaned encoder produces highly similar embeddings for samples with trigger while a clean encoder does not. Table 2 shows that, in a clean encoder, the trigger can increase the cosine similarity of samples from 0.2193 to 0.2922 (in the first row). The increase is limited and insignificant. As shown in Figure. 2b, the clean encoder can still correctly separate inputs with the trigger. In contrast, as the backdoor attack forces the samples with trigger to be close to the attack target, it creates a *dense area* (shown in Figure. 2d) in a backdoored encoder where embeddings share a high similarity (0.9904).

Table 1. Limitations of Existing Backdoor Scanners When Scanning Encoders. The encoder is pre-trained on CIFAR-10. The downstream classifiers (SVHN, STL-10, and GTSRB) are trained for 500 epochs. The attack target is an image of label *one* from SVHN. NC Anomaly Index > 2.0 and ABS REASR > 0.88 indicate the classifier is considered trojaned.

Downsteam Task	Classifier Performance			Neural Cleanse		ABS	
	Accuracy	ASR	Training Time (m)	Anomaly Index	Detection Time (m)	REASR	Detection Time (m)
SVHN	0.69	1.00	60.41	2.18	5.36	1.00	2.96
STL-10	0.76	-	14.61	1.23	4.54	0.44	2.89
GTSRB	0.82	-	37.07	1.49	16.09	0.36	3.13

Table 2. Cosine similarity within 1024 random CIFAR10 images. Both clean and backdoored encoders are pretrained on CIFAR10.

	Samples w/o Trigger	Samples w/ Trigger
Clean Encoder	0.2193	0.2922
Backdoored Encoder	0.2442	0.9904



(a) Clean Encoder + Clean Input (b) Clean Encoder + Trojaned Input

(c) Trojaned Encoder + Clean Input (d) Trojaned Encoder + Trojaned Input

Figure 2. Embedding Space Distributions. Subfigures (a) and (b) are for a clean encoder and (c), (d) for a trojaned encoder; colors denote class labels; faded colors in (b) and (d) denote embeddings of clean samples. For the clean encoder, even if the inputs are stamped with the ground truth trigger, the embeddings are well separable in (b). However in (d), a trojaned encoder produces similar embeddings for trojaned inputs.

Observation III: Compared to clean encoders, backdoored encoders need much smaller perturbations to cause samples to fall into the dense area. Figure. 2d illustrates that the dense area (of a trojaned encoder) is surrounded by and close to clusters of clean samples. However, in the clean encoder, larger perturbations are required to induce highly similar embeddings for input samples, as the clean encoder produces more scattered embeddings.

Intuitions. The dense area is where the attack target lies. This is analogous to the target label of backdoor attacks in supervised learning. The key difference is that, in the supervised learning setting, backdoor scanners can scan each label and then identify the most suspicious label as the target. However in SSL, there are no labels for scanners to iterate over. As such, existing backdoor scanners cannot be applied to determine whether a model is backdoored in SSL.

To overcome the challenge, our design aims to decide whether there exists a central dense area in the encoder's embedding space (surrounded by the embeddings of clean samples). Intuitively, a backdoored encoder with a central dense area only needs a small perturbation to push clean samples to the dense region. A clean encoder, on the other hand, does not have such a dense area, meaning that high similarity among embeddings cannot be easily achieved by stamping a small trigger on samples. Our technique hence detects backdoors at the encoder level, without the need of a target label. We elaborate design details in the rest of the section.

4.2. Methodology

Trigger inversion is one of widely used techniques in backdoor scanning [14,31,45,50,51]. It works by optimizing a trigger pattern, which can induce the targeted misclassification while having a small trigger size. Existing trigger inversion was originally designed for supervised learning scenarios, where there are explicit labels. The size of trigger can be used as a metric to quantify the distance between the target label and other non-target labels. In SSL, however, no explicit label exists. Existing trigger inversion is not able to optimize or update the pattern towards some intended objective (a target label). Inspired by the above observations, we propose to find the aforementioned dense area with only a small trigger. It can be formulated as a constrained optimization problem. With the constraint that samples stamped with the same trigger must have similar embeddings, the trigger size shall be optimized to the minimal.

Figure. 3 shows the overview of our technique. A randomly initialized trigger and a shadow dataset (e.g., a subset of pre-training dataset) are fed into the subject encoder to

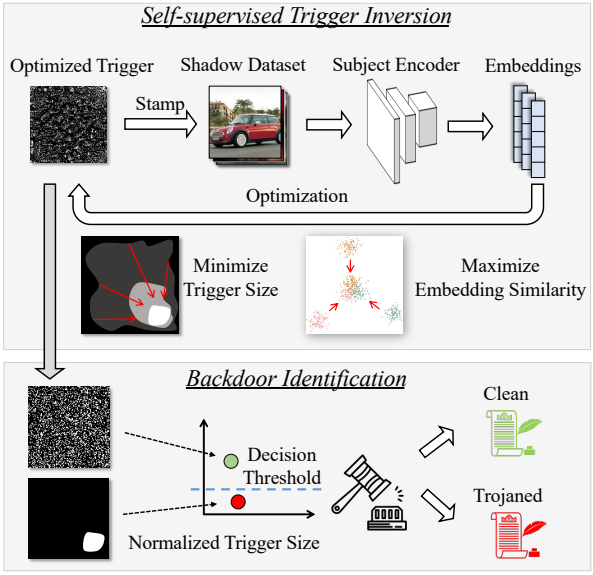


Figure 3. DECREE Overview

compute embeddings. The cosine similarity of these embeddings guides the optimization of trigger. With the constraint that samples stamped with the same trigger must have similar embeddings, the trigger size is iteratively optimized to the minimal. The optimized trigger is used for calculating a metric that gauges the normalized trigger size. The metric is then used to determine if the encoder is trojaned.

In Section 4.2.1, we explain the details of our self-supervised trigger inversion. In Section 4.2.2, we demonstrate how to use inverted triggers to conduct encoder-level backdoor detection.

4.2.1 Self-supervised Trigger Inversion

To generate a trigger that can induce intended backdoor behavior in an encoder, we use two trainable variables, a mask and a pattern, to denote the trigger pattern. Specifically, the mask is utilized to indicate how much a pixel on the input image is replaced by the pattern. We use the following equation to formalize the trigger injection:

$$\mathcal{F}(\boldsymbol{x}, \boldsymbol{m}, \boldsymbol{t}) = \boldsymbol{x}', \tag{1}$$

$$\boldsymbol{x}'_{i,j,c} = \boldsymbol{m}_{i,j} \cdot \boldsymbol{x}_{i,j,c} + (1 - \boldsymbol{m}_{i,j}) \cdot \boldsymbol{t}_{i,j,c}, \ \forall i \in H, j \in W, c \in C. \tag{2}$$

$\mathcal{F}$ denotes a function that stamps trigger pattern $\boldsymbol{t}$ onto an input image $\boldsymbol{x}$ and outputs an backdooored image $\boldsymbol{x}'$; $\boldsymbol{m}$ is a mask indicates how much the original pixel values are retained. It has continuous values ranging from 0 to 1. The input image has three dimensions, namely, height H, width W, and channel C; $\boldsymbol{x}_{i,j,c}$ refers to the pixel value of image $\boldsymbol{x}$ at height i, width j, and channel c. Note that $\boldsymbol{m}_{i,j}$ only has two dimensions as the mask is applied on a pixel in a way ranging from replacing it ($\boldsymbol{m}_{i,j} = 0$) to retaining it ($\boldsymbol{m}_{i,j} = 1$), regardless of the color channels.

The goal of self-supervised trigger inversion is to optimize a trigger such that clean samples stamped with the trigger have highly similar embeddings. In SSL, the cosine similarity is commonly used as a metric to denote the distance between a pair of inputs [5, 6, 40]. We leverage the same metric to measure how close the embeddings of trigger-stamped samples are. Formally, given two inputs $\boldsymbol{x}_p$ and $\boldsymbol{x}_q$ from a dataset $\mathcal{D} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N\}$, we have:

$$\mathcal{L}_{p,q}(E, \boldsymbol{m}, \boldsymbol{t}) = -cos\Big(E\big(\mathcal{F}(\boldsymbol{x}_p, \boldsymbol{m}, \boldsymbol{t})\big), E\big(\mathcal{F}(\boldsymbol{x}_q, \boldsymbol{m}, \boldsymbol{t})\big)\Big). \tag{3}$$

E is the subject encoder, and $\boldsymbol{m}$ and $\boldsymbol{t}$ are the trigger variables discussed in Eq. 1 and Eq. 2. The two inputs $\boldsymbol{x}_p$ and $\boldsymbol{x}_q$ are transformed by function $\mathcal{F}$ in Eq. 1 to stamp the trigger. To achieve high similarities among samples that approximate the search for the dense area in the embedding space of encoder, DECREE samples a batch of inputs to stabilize the search process. The average of pair-wise similarity within a batch is computed as follows:

$$\mathcal{L} = \frac{1}{N^2} \sum_{p=1}^{N} \sum_{q=1}^{N} \mathcal{L}_{p,q}(E, \boldsymbol{m}, \boldsymbol{t}), \tag{4}$$

where N is the batch size; $\mathcal{L}$ is used as the constraint during optimization, assuring that the samples stamped with the optimized trigger are in the dense area in embedding space.

We then leverage *Observation III* in Section 4.1 and minimize the size of the trigger. We use the L^1 norm to quantify the mask size. Self-supervised trigger inversion is formulated as the following constrained optimization problem.

$$\min \|\boldsymbol{m}\|_1, s.t.\ \mathcal{L} < \beta. \tag{5}$$

β is a threshold assuring the average similarity is high.

During trigger inversion, a set of clean samples are needed for the optimization. As we do not assume the knowledge of any downstream tasks like what existing backdoor scanners do, we leverage the pre-training dataset that is used for constructing the encoder. It is impractical to have the whole pre-training dataset as one can directly train a new clean encoder on it without the need of scanning the given encoder. We hence only assume a small subset of the pre-training dataset ($< 10\%$) for trigger inversion. In extreme cases, the pre-training dataset of the given encoder may not be publicly available. We resort to leveraging an external dataset, called *shadow dataset*, for trigger inversion and backdoor scanning. Since the pre-training set and downstream datasets do not share the same set of samples or are even from different data distributions, the attack effectiveness solely depends on building a strong connection between the injected trigger and the target embedding. It hence does not matter what data are used for trigger inversion. Our results in Section 5.5 demonstrate that DECREE can indeed effectively detect backdoored encoders using a shadow dataset.

4.2.2 Backdoor Identification

Recall that a challenge of detecting backdoors in SSL is that there are no labels. Therefore existing backdoor scanners cannot identify the potential target label, which is the key to determining whether a model is backdoored in supervised learning. To overcome this challenge, DECREE introduces a new metric $\mathcal{PL}^n$:

$$\mathcal{PL}^n(E) = \frac{\|\widetilde{\boldsymbol{m}}\|_n}{\|\hat{x}\|_n}. \tag{6}$$

$\|\cdot\|_n$ denotes the L^n norm of a vector; $\widetilde{\boldsymbol{m}}$ denotes the trigger inverted from a given encoder E; and $\hat{x}$ denotes the input sample that has the maximum L^n norm in the input space of that encoder. $\mathcal{PL}^n(E)$, denoting the *Proportionate-L^n Norm* of an encoder E, is thus defined as the ratio of the inverted trigger's L^n norm to the maximum L^n norm of the encoder's input space. Note that $\mathcal{PL}^n$ is an encoder-level metric, approximating the distance from clean samples to the dense area. In this way, DECREE does not need to identify the target label.

As discussed in Section 4.1, triggers inverted from backdoored encoders shall be smaller than those from clean encoders. Thus for a backdoored encoder, DECREE has a better chance to invert a small trigger that can induce the encoder to output two similar embeddings for two dissimilar inputs. Based on the proposed $\mathcal{PL}^n$ and the above intuition, DECREE uses the following formula to identify backdoors in encoders.

$$\widetilde{P(E)} = \mathbb{B}\Big(\mathcal{PL}^1(E), \tau\Big). \tag{7}$$

$\widetilde{P(E)}$ is the estimated probability that a given encoder E contains a backdoor. $\mathbb{B}$ is a binary step function that returns 1 if its first parameter is less than a given threshold τ and 0 otherwise. Essentially, if the inverted trigger of a given encoder only occupies a small part of the input data sample, we consider the encoder is very likely a trojaned encoder.

5. Evaluation

We use the following research questions (RQs) to evaluate DECREE:

RQ1: How effective is our method?

RQ2: How efficient is our method?

RQ3: How robust is our method against adaptive attack?

RQ4: How effective is our method if the defender has no access to the pre-training dataset?

5.1. Experiment Setup

We employ five commonly used datasets, CIFAR10 [22], GTSRB [17], SVHN [36], STL-10 [11], and ImageNet [41], for pre-training encoders and training downstream classifiers. We use three well-known model architectures, ResNet18, ResNet34, and ResNet50 [16]. As the CLIP dataset [40] is not publicly available, we downloaded a pre-trained encoder from [39] and use ImageNet to finetune the encoder by applying SimCLR [5] algorithm.

For backdoor attacks, we consider three categories in the SSL setting, namely *Image-on-Image*, *Image-on-Pair*, and *Text-on-Pair*, as discussed in Section 2.3. Note that there are only a limited number of public backdoored encoders, we hence use the official implementation [20] or implement the attacks strictly following the original paper [3] to construct backdoored encoders. For *Image-on-Image* and *Image-on-Pair* attacks, we choose a "priority" image from GTSRB, a "one" image from SVHN, and a "truck" image from STL-10 as attack targets. We only consider backdoored encoders that achieve at least 99% attack success rate in the targeted downstream classifiers. For *Text-on-Pair* attack, we choose the label text "priority" for GTSRB, "one" for SVHN, and "truck" for STL-10 to fill in a prompt list (shown in Table 6 in Appendix B) and use these text captions as attack targets. The z-score introduced in [3] quantifies to what extent the subject encoder is trojaned. We only consider backdoored encoders with a z-score greater than 2.5 for evaluation. We set $\beta = -0.99$ and $\tau = 0.1$ during the detection. We use 444 encoders (111 benign and 333 backdoored) to evaluate DECREE. Details are shown in Appendix A.

5.2. RQ1: Effectiveness of Our Method

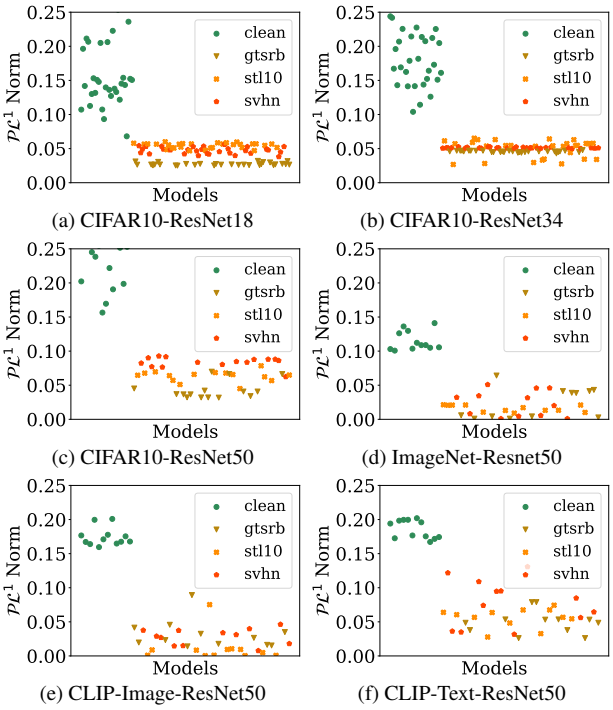
We evaluate the performance of DECREE by using common metrics (e.g., detection accuracy, ROC-AUC). We also show the distributions of inverted triggers for clean and backdoored encoders and study how the two sets are separated by DECREE.

The detection results of DECREE are shown in Table 3. We evaluate on three attack categories, namely *Image-on-Image*, *Image-on-Pair*, and *Text-on-Pair*. For each attack category, we choose three attack targets, from GTSRB, SVHN and STL-10 respectively.

Observe that DECREE can effectively detect almost all the backdoored encoders with more than 95% accuracy in most cases. Particularly, for 14 out of 18 scenarios, DECREE has 100% detection accuracy. For *Text-on-Pair* on SVHN, the detection accuracy is slightly lower (87.5%). This is because the attack targets for this case are natural language sentences, and they usually have multiple target instances. For example, a trigger with the label text "truck" can use both "a picture of truck" and "a nice photo of truck" as attack targets, making the triggers less centralized than those attacks on images. Note that we use the same threshold for all the application/attack settings. That said, with the knowledge of the particular application scenario (*Text-on-Pair*), DECREE can still effectively distinguish backdoored encoders from clean encoders by slightly increasing the threshold, as depicted in Figure 4f. The last row in Table 3 show the summarized performance. We can see that

Table 3. Detection Performance. The first three columns list the attack category, the dataset used for pre-training encoders, and the model architecture. RN18, RN34, and RN50 denote model architecture ResNet18, ResNet34, and ResNet50, respectively. The following three column blocks present the results for which dataset the attack target comes from, i.e., GTSRB, SVHN, and STL-10. Columns in each block show the number of true positives (TP), false positives (FP), false negatives (FN), true negatives (TN) when we use DECREE to detect backdoored encoders. Acc denotes the overall detection accuracy.

Attack Category	Pre-training Dataset	Model Arch	GTSRB atk					SVHN atk					STL-10 atk				
			TP	FP	FN	TN	Acc	TP	FP	FN	TN	Acc	TP	FP	FN	TN	Acc
Img-on-Img	CIFAR10	RN18	30	2	0	28	96.7	30	2	0	28	96.7	30	2	0	28	96.7
		RN34	30	0	0	30	100	30	0	0	30	100	30	0	0	30	100
		RN50	15	0	0	15	100	15	0	0	15	100	15	0	0	15	100
	ImageNet	RN50	12	0	0	12	100	12	0	0	12	100	12	0	0	12	100
Img-on-Pair	CLIP	RN50	12	0	0	12	100	12	0	0	12	100	12	0	0	12	100
Text-on-Pair	CLIP	RN50	12	0	0	12	100	9	0	3	12	87.5	12	0	0	12	100
Summary	-	-	111	2	0	109	**99.1**	108	2	3	109	**97.7**	111	2	0	109	**99.1**



(a) CIFAR10-ResNet18 (b) CIFAR10-ResNet34 (c) CIFAR10-ResNet50 (d) ImageNet-Resnet50 (e) CLIP-Image-ResNet50 (f) CLIP-Text-ResNet50

Figure 4. Distribution of Inverted Triggers. Each sub-figure corresponds to one setting (one line in Table 3) and depicts the results for that setting. The x-axis denotes different models and the y-axis denotes the $\mathcal{PL}^1$-Norm value. The green markers denote inverted triggers for clean encoders while other color markers (i.e., brown, orange, and red markers) denote inverted triggers for backdoored encoders with attack targets coming from GTSRB, STL-10 and SVHN, respectively.

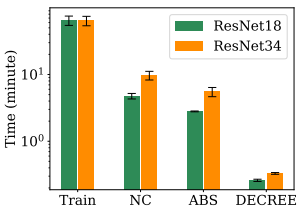


Figure 5. Time Efficiency

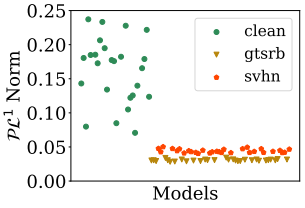


Figure 6. Performance without Access to Pre-training Dataset

DECREE achieves a detection accuracy of near 100% in all cases on average, delineating its effectiveness. We also use the ROC (Receiver Operating Characteristic) curve to study the relation between true positive rate and false positive rate as shown in Figure 8 in Appendix D.

We study the distributions of inverted triggers for clean and backdoored encoders, which are shown in Figure 4. Each sub-figure corresponds to one setting (one line in Table 3) and depicts the results for that setting. Observe that in all scenarios, inverted triggers for backdoored encoders have smaller $\mathcal{PL}^1$-Norm than those for clean encoders. The triggers for backdoored encoders tend to cluster in small $\mathcal{PL}^1$-Norm values (< 0.1). This demonstrates the reason why DECREE is able to effectively detect backdoored encoders with a same threshold. We also visually show that the inverted triggers for backdoored encoders have much fewer perturbed pixels compared to those for clean encoders. Please see detailed results and discussion in Appendix C.

5.3. RQ2: Efficiency of Our Method

In this section, we evaluate the efficiency of DECREE in comparison with two SOTA backdoor scanning techniques, i.e., Neural Cleanse (NC) [50] and ABS [31]. Recall in Section 3, we observe that existing detection methods need the knowledge of downstream tasks. In addition, they also require samples from the downstream dataset for detection. For a fair comparison, we assume existing detectors have full access to the downstream dataset, with which they can train a corresponding downstream classifier and perform the detection based on the classifier and downstream task samples.

We conduct experiments on 10 backdoored encoders trained on CIFAR-10 with ResNet18 and ResNet34 architectures. The attack target is a "one" image from the SVHN dataset. Figure 5 shows the results. As existing techniques need to train downstream classifiers, we also show the training time of classifiers in the first two columns.

Observe that training a classifier takes a large amount of time, more than 1 hour. The runtime of existing techniques is around 2-10 minutes. DECREE, on the other hand, only takes 15-20 seconds. It is 6-30 times faster than baselines, even without considering the training time for downstream classifiers. This is because DECREE generates just one trigger for each encoder and do not have to scan each label like what existing methods do.

5.4. RQ3: Adaptive Attack

We consider a stronger attack that aims to evade the detection of DECREE with the full knowledge of our detection pipeline. Assume the loss function used in the original attack is $\mathcal{L}_{atk}$. The stronger attack also considers $\mathcal{L}_{sim}$, the same as $\mathcal{L}$ in Eq. 4. $\mathcal{L}_{sim}$ quantifies the similarity among inputs stamped with the trigger. The attacker aims to enlarge this loss to make those samples less similar. Therefore, the objective of the adaptive attack is as follows.

$$\arg\min \mathcal{L}_{adapt} = \mathcal{L}_{atk} - \alpha \cdot \mathcal{L}_{sim}, \quad \alpha > 0 \qquad (8)$$

We conduct experiments on a ResNet18 encoder trained on CIFAR10 and the attack target is a "one" image from the SVHN dataset. We set $\alpha = 1$. The adaptive attack can produce a trojaned encoder that has an inverted trigger with a $\mathcal{PL}^1$-Norm of 0.14 , evading DECREE's detection. However, the ASR on downstream STL-10 degrades from 99.9% to 69.9%. Intuitively, $\mathcal{L}_{adapt}$ forces the embeddings of inputs stamped with the trigger to have a similar embedding with the attack target while trying to make them orthogonal to each other. It hence is difficult for the attack to achieve a high ASR and evade our detection (i.e. inputs stamped with triggers share high cosine similarity) at the same time. Please see more details in Appendix F.

5.5. RQ4: No Access to Pre-training Dataset

In previous experiments, we use a small subset of the pre-training dataset for trigger inversion. In extreme cases, the pre-training dataset may not be available, which significantly increases the difficulty of backdoor scanning. We evaluate DECREE in this setting to show its robustness. We use CIFAR10 as the pre-training dataset, GTSRB and SVHN as origins of attack targets, and STL-10 as the shadow dataset for detection. As shown in Figure 6, the distribution of inverted triggers in this setting is similar to those in Figure 4. DECREE can clearly separate clean and backdoored encoders based on $\mathcal{PL}^1$-Norm, delineating the generalizability of DECREE. Note that CLIP pre-training dataset is not public. Rows *Image-on-Pair* and *Text-on-Pair* in Table 3 also fall into this challenging threat model. Figure 4e and Figure 4f show the detection results for these two.

One key factor contributing to the generalizability of DECREE is that encoders pre-trained on unlabeled data via contrastive learning do not easily overfit on a certain dataset. In addition, $\mathcal{PL}^1$-Norm considers different input dimensions so that DECREE is insensitive to different attack settings.

5.6. Other Experiments

Ablation Study. We conduct ablation studies to validate the robustness of DECREE against various trigger configurations (e.g., color, size, texture) and different attack strategies. Details are shown in Appendix G.1. We also study the hyperparameters (shadow dataset size M and decision threshold τ) and show the performance is insensitive to different hyperparameters. Details can be found in Appendix G.2.

Advanced Attacks. We adapt 2 dynamic attacks [28, 38] from supervised learning into our settings and find that such attacks can hardly succeed in SSL setting. Details can be found in Appendix H.

More SSL Attacks. We also study 3 emerging attacks [26, 43, 57]. We find that DECREE can detect acute attacks (i.e., high ASR) with patch-like triggers [57], but may fail on attacks with pervasive triggers [26] or stealthy attacks [43]. Details can be found in Appendix I.

6. Conclusion

We propose the first backdoor detection method DECREE for pre-trained encoders. Our method fills in the gap where existing detection techniques only focus on supervised learning scenarios. Our evaluation shows that DECREE can effectively and efficiently separate benign and trojaned encoders. Our method is also robust against adaptive attacks and generalizes to a more challenging threat model.

Limitation of Our Work. We currently do not handle text-format trigger. Our method mainly focuses on three types of attacks (*Image-on-Image*, *Image-on-Pair*, and *Text-on-Pair*), the attack subject of which is an image encoder. For *Text-on-Text* attack, it introduces extra challenges to invert text-format triggers as the input in NLP is discrete (e.g., words), different from the pixel values in computer vision.

Acknowledgement

We thank the anonymous reviewers for their constructive comments. This research was supported, in part by IARPA TrojAI W911NF-19-S-0012, NSF 1901242 and 1910300, ONR N000141712045, N000141410468 and N000141712947. Any opinions, findings, and conclusions in this paper are those of the authors only and do not necessarily reflect the views of our sponsors.

References

[1] Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes, 2016. 1

[2] Eitan Borgnia, Valeriia Cherepanova, Liam Fowl, Amin Ghiasi, Jonas Geiping, Micah Goldblum, Tom Goldstein, and Arjun Gupta. Strong data augmentation sanitizes poisoning and backdoor attacks without an accuracy tradeoff. *arXiv preprint arXiv:2011.09527*, 2020. 2

[3] Nicholas Carlini and Andreas Terzis. Poisoning and backdooring contrastive learning. In *International Conference on Learning Representations*, 2022. 1, 2, 3, 6, 12, 15

[4] Ming-Wei Chang, Lev-Arie Ratinov, Dan Roth, and Vivek Srikumar. Importance of semantic representation: Dataless classification. In *Aaai*, volume 2, pages 830–835, 2008. 1

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. 1, 2, 5, 6

[6] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*, 2020. 2, 5

[7] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 3

[8] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017. 1, 2

[9] Siyuan Cheng, Yingqi Liu, Shiqing Ma, and Xiangyu Zhang. Deep feature space trojan attack of neural networks by controlled detoxification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1148–1156, 2021. 2

[10] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546. IEEE, 2005. 1

[11] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011. 3, 6

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 1, 2

[13] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 2019. 1, 2

[14] Wenbo Guo, Lun Wang, Xinyu Xing, Min Du, and Dawn Song. Tabor: A highly accurate approach to inspecting and restoring trojan backdoors in ai systems. *arXiv preprint arXiv:1908.01763*, 2019. 2, 4

[15] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 1, 2

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[17] Sebastian Houben, Johannes Stallkamp, Jan Salmen, Marc Schlipsing, and Christian Igel. Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark. In *International Joint Conference on Neural Networks*, number 1288, 2013. 3, 6

[18] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916. PMLR, 2021. 1, 3

[19] Jinyuan Jia, Xiaoyu Cao, and Neil Zhenqiang Gong. Certified robustness of nearest neighbors against data poisoning attacks. In *AAAI Conference on Artificial Intelligence*, 2020. 2

[20] Jinyuan Jia, Yupei Liu, and Neil Zhenqiang Gong. BadEncoder: Backdoor attacks to pre-trained encoders in self-supervised learning. In *IEEE Symposium on Security and Privacy*, 2022. 1, 2, 3, 6, 12, 15

[21] Soheil Kolouri, Aniruddha Saha, Hamed Pirsiavash, and Heiko Hoffmann. Universal litmus patterns: Revealing backdoor attacks in cnns. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 301–310, 2020. 2

[22] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 3, 6

[23] Keita Kurita, Paul Michel, and Graham Neubig. Weight poisoning attacks on pre-trained models. In *58th Annual Meeting of the Association for Computational Linguistics*, 2020. 3

[24] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *2009 IEEE conference on computer vision and pattern recognition*, pages 951–958. IEEE, 2009. 1

[25] Hugo Larochelle, Dumitru Erhan, and Yoshua Bengio. Zero-data learning of new tasks. In *AAAI*, volume 1, page 3, 2008. 1

[26] Changjiang Li, Ren Pang, Zhaohan Xi, Tianyu Du, Shouling Ji, Yuan Yao, and Ting Wang. Demystifying self-supervised trojan attacks, 2022. 8, 15

[27] Yige Li, Nodens Koren, Lingjuan Lyu, Xixiang Lyu, Bo Li, and Xingjun Ma. Neural attention distillation: Erasing backdoor triggers from deep neural networks. In *International Conference on Learning Representations*, 2021. 2

[28] Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu. Invisible backdoor attack with sample-specific triggers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16463–16472, October 2021. 2, 8, 15

[29] Junyu Lin, Lei Xu, Yingqi Liu, and Xiangyu Zhang. Composite backdoor attack for deep neural network by mixing

existing benign features. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 113–131, 2020. 2

[30] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *International Symposium on Research in Attacks, Intrusions, and Defenses*, pages 273–294. Springer, 2018. 2

[31] Yingqi Liu, Wen-Chuan Lee, Guanhong Tao, Shiqing Ma, Yousra Aafer, and Xiangyu Zhang. Abs: Scanning neural networks for back-doors by artificial brain stimulation. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 1265–1282, 2019. 2, 3, 4, 7, 14

[32] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. In *NDSS*, 2018. 1, 2, 15

[33] Yunfei Liu, Xingjun Ma, James Bailey, and Feng Lu. Reflection backdoor: A natural backdoor attack on deep neural networks. In *European Conference on Computer Vision*, pages 182–199. Springer, 2020. 2

[34] Yingqi Liu, Guangyu Shen, Guanhong Tao, Zhenting Wang, Shiqing Ma, and Xiangyu Zhang. Ex-ray: Distinguishing injected backdoor from natural features in neural networks by examining differential feature symmetry. *arXiv preprint arXiv:2103.08820*, 2021. 2

[35] Michael McCoyd, Won Park, Steven Chen, Neil Shah, Ryan Roggenkemper, Minjune Hwang, Jason Xinyu Liu, and David Wagner. Minority reports defense: Defending against adversarial patches. In *International Conference on Applied Cryptography and Network Security*, pages 564–582. Springer, 2020. 2

[36] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011. 3, 6

[37] Tuan Anh Nguyen and Anh Tran. Input-aware dynamic backdoor attack. *Advances in Neural Information Processing Systems*, 33, 2020. 2

[38] Tuan Anh Nguyen and Anh Tuan Tran. Wanet-imperceptible warping-based backdoor attack. In *International Conference on Learning Representations*, 2020. 2, 8, 15

[39] OpenAI. https://github.com/openai/clip, 2021. 6

[40] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 2, 3, 5, 6

[41] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 6

[42] Aniruddha Saha, Akshayvarun Subramanya, and Hamed Pirsiavash. Hidden trigger backdoor attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11957–11965, 2020. 2

[43] Aniruddha Saha, Ajinkya Tejankar, Soroush Abbasi Koohpayegani, and Hamed Pirsiavash. Backdoor attacks on self-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13337–13346, June 2022. 3, 8, 15

[44] Ahmed Salem, Rui Wen, Michael Backes, Shiqing Ma, and Yang Zhang. Dynamic backdoor attacks against machine learning models. *arXiv preprint arXiv:2003.03675*, 2020. 2

[45] Guangyu Shen, Yingqi Liu, Guanhong Tao, Shengwei An, Qiuling Xu, Siyuan Cheng, Shiqing Ma, and Xiangyu Zhang. Backdoor scanning for deep neural networks through k-arm optimization. In *International Conference on Machine Learning*, 2021. 2, 4

[46] Lujia Shen, Shouling Ji, Xuhong Zhang, Jinfeng Li, Jing Chen, Jie Shi, Chengfang Fang, Jianwei Yin, and Ting Wang. Backdoor pre-trained models can transfer to all. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. ACM, nov 2021. 1, 3

[47] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. *Advances in neural information processing systems*, 29, 2016. 3

[48] Guanhong Tao, Yingqi Liu, Guangyu Shen, Qiuling Xu, Shengwei An, Zhuo Zhang, and Xiangyu Zhang. Model orthogonalization: Class distance hardening in neural networks for better security. In *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2022. 2

[49] Guanhong Tao, Guangyu Shen, Yingqi Liu, Shengwei An, Qiuling Xu, Shiqing Ma, Pan Li, and Xiangyu Zhang. Better trigger inversion optimization in backdoor scanning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13368–13378, 2022. 2

[50] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 707–723. IEEE, 2019. 2, 3, 4, 7, 14

[51] Ren Wang, Gaoyuan Zhang, Sijia Liu, Pin-Yu Chen, Jinjun Xiong, and Meng Wang. Practical detection of trojan neural networks: Data-limited and data-free cases. In *European Conference on Computer Vision*, pages 222–238. Springer, 2020. 2, 4

[52] Dongxian Wu and Yisen Wang. Adversarial neuron pruning purifies backdoored deep models. *Advances in Neural Information Processing Systems*, 34, 2021. 2

[53] Chong Xiang, Arjun Nitin Bhagoji, Vikash Sehwag, and Prateek Mittal. Patchguard: A provably robust defense against adversarial patches via small receptive fields and masking. In *30th USENIX Security Symposium (USENIX Security 21)*, 2021. 2

[54] Xiaojun Xu, Qi Wang, Huichen Li, Nikita Borisov, Carl A. Gunter, and Bo Li. Detecting ai trojans using meta neural analysis. *2021 IEEE Symposium on Security and Privacy (SP)*, pages 103–120, 2021. 2, 3

[55] Yuanshun Yao, Huiying Li, Haitao Zheng, and Ben Y Zhao. Latent backdoor attacks on deep neural networks. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 2041–2055, 2019. 2

[56] Yi Zeng, Han Qiu, Shangwei Guo, Tianwei Zhang, Meikang Qiu, and Bhavani Thuraisingham. Deepsweep: An evaluation framework for mitigating dnn backdoor attacks using data augmentation. *arXiv preprint arXiv:2012.07006*, 2020. 2

[57] Jinghuai Zhang, Hongbin Liu, Jinyuan Jia, and Neil Zhenqiang Gong. Corruptencoder: Data poisoning based backdoor attacks to contrastive learning, 2023. 8, 15

[58] Kaiyuan Zhang, Guanhong Tao, Qiuling Xu, Siyuan Cheng, Shengwei An, Yingqi Liu, Shiwei Feng, Guangyu Shen, Pin-Yu Chen, Shiqing Ma, and Xiangyu Zhang. FLIP: A provable defense framework for backdoor mitigation in federated learning. In *The Eleventh International Conference on Learning Representations*, 2023. 2

[59] Pu Zhao, Pin-Yu Chen, Payel Das, Karthikeyan Natesan Ramamurthy, and Xue Lin. Bridging mode connectivity in loss landscapes and adversarial robustness. In *International Conference on Learning Representations (ICLR 2020)*, 2020. 2

Appendix

We provide a table of contents below for better navigation of the appendix.

Appendix A provides details of evaluation setup.
Appendix B introduces the settings of backdoor attacks on self-supervised learning that are adopted in our evaluation.
Appendix C studies the triggers inverted by DECREE.
Appendix D uses ROC curve to quantify the effectiveness of DECREE.
Appendix E evaluates the efficiency of DECREE in comparison with two SOTA backdoor scanning techniques.
Appendix F designs an adaptive attack aiming to evade our detection.
Appendix G.1 studies the effectiveness of DECREE against different trigger patterns and sizes.
Appendix G.2 shows the effectiveness of threshold τ.
Appendix H explores the feasibility of adapting 2 existing advanced attacks from supervised learning into self-supervised learning setting.
Appendix I discusses on 3 emerging SSL backdoor attacks.

A. Evaluation Setup

Table 4 shows the statistics of evaluated attacks, datasets, and encoders. Column 1 denotes the attack category. Column 2 shows the pre-training datasets used for constructing encoders. Columns 3-5 present the model architecture, input image shape, and the number of (trainable) model parameters. Column 6 shows the number of clean encoders for each setting. For backdoored encoders, we choose one label from each *attack datasets* as attack target label. For example, when attack dataset is GTSRB, we choose a "priority" image as attack target in *Image-on-Image* and *Image-on-Pair* settings and choose the word "priority" to fill in prompts in *Text-on-Pair* setting. We introduce more details in Appendix B. We evaluate on three attack datasets that are shown in Columns 7-9. The numbers denote how many backdoored encoders are trained for the corresponding attack datasets. In total, we have 444 encoders (111 benign and 333 backdoored).

B. Attack Settings

B.1. Image-on-Image & Image-on-Pair

For *Image-on-Image* and *Image-on-Pair* attacks, we follow the code released by BadEncoder [20] to construct backdoored encoders. Specifically, the main idea is that, given a clean encoder E, the attacker aims to get a trojaned encoder E' such that E and E' satisfy the following 3 properties: (1) For each clean input image x, $E(x)$ and $E'(x)$ should be similar. (2) For the target image r, $E(r)$ and $E'(r)$ should be similar. (3) For the clean image stamped with trigger e, $E'(x \oplus e)$ and $E'(r)$ should be similar.

For each attack datasets, we use the same target images as [20]. We select trojaned encoders that can train downstream classifiers with ASR $> 99\%$ and accuracy $> 70\%$.

B.2. Text-on-Pair

For *Text-on-Pair* attack, we follow the method introduced in [3]. The main idea is to construct a malicious training dataset $\mathcal{P}$ (size of which is a small fraction of pre-training dataset size). $\mathcal{P}$ is defined as $\mathcal{P} = \{(x_i \oplus e, c)\}_i$, where x_i are clean images, e is trigger and c is attack target caption. The caption is formed by filling in prompts (shown in Table 6) with a word of interest from attack datasets(shown in Table 5). We choose backdoored encoders with z-score [3] higher than 2.5.

C. Triggers Inverted by DECREE

In Figure 7, we show the triggers inverted by DECREE. The ground truth trigger is a white square located at the right bottom of the image. For Figure 4a 4b 4c, the ground truth trigger shape (height, width, channel) is (10, 10, 3). For Figure 4d 4e 4f, the ground truth trigger shape (height, width, channel) is (24, 24, 3).

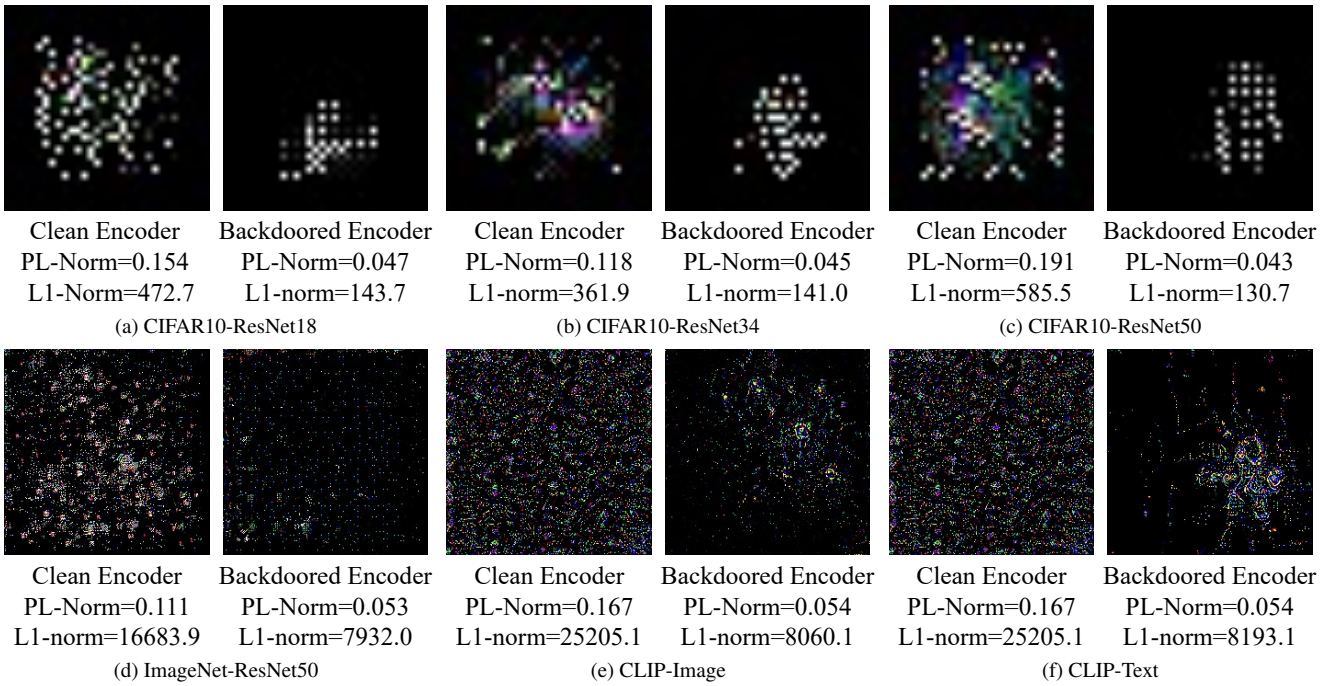
For each setup, we show a trigger inverted from clean encoder, and a trigger inverted from backdoored encoder. We also report the value of $\mathcal{PL}^1$-Norm for each trigger in the figure. Notice that (1) triggers inverted from backdoored encoders exploit significantly less pixels than those inverted from clean encoders, and thus their $\mathcal{PL}^1$-Norm are lower, (2) triggers inverted from backdoored encoders tend to cluster and shift towards the corner, while those inverted from clean encoders are likely to evenly distribute throughout the entire image. For example, in Figure 7a, the trigger from clean encoder scatters over almost the whole image, while the trigger from the backdoored encoder centralizes at the lower right part of the image. One can still make similar observations under *Text-on-Pair* attack. Take Figure 7f as an example. The trigger from clean encoder evenly distributes across the image, while the trigger from backdoored encoder densely distributes in the lower right region.

D. ROC of DECREE on Different Datasets

We further use the ROC (Receiver Operating Characteristic) to quantify the effectiveness of our detection method. Given a set of encoders, DECREE inverts triggers from each of them and computes $\mathcal{PL}^1$-Norm. After that, to distinguish the backdoored encoders from the benign ones, one can set a threshold for $\mathcal{PL}^1$-Norm. The ROC curves are shown in Figure 8. These curves depict how the True Positive Rate (TPR, marked by the vertical axis) and False Positive Rate (FPR, marked by the horizontal axis) change when different thresholds are selected. The green curve denotes the ROC obtained on all the 444 encoders. That is, we set one univer-

Table 4. Model Statistics

Attack Category	Pre-training Dataset	Model Arch	Input Size	#Params	Clean Encoder	Attack Datasets		
						GTSRB	SVHN	STL-10
Image-on-Image	CIFAR10	ResNet18	32×32×3	11,168,832	30	30	30	30
		ResNet34	32×32×3	21,276,992	30	30	30	30
		ResNet50	32×32×3	23,500,352	15	15	15	15
	ImageNet	ResNet50	224×224×3	25,557,032	12	12	12	12
Image-on-Pair	CLIP Dataset	ResNet50	224×224×3	38,316,896	12	12	12	12
Text-on-Pair	CLIP Dataset	ResNet50	224×224×3	38,316,896	12	12	12	12



(a) CIFAR10-ResNet18 (b) CIFAR10-ResNet34 (c) CIFAR10-ResNet50

(d) ImageNet-ResNet50 (e) CLIP-Image (f) CLIP-Text

Figure 7. Inverted Triggers. Subfigures 4a 4b 4c 4d are *Image-on-Image* attacks. Subfigure 4e is *Image-on-Pair* attack. Subfigure 4f is *Text-on-Pair* attack. Note that our goal is to do detection and thus it is not that necessary to invert exactly the same trigger as the injected one. DECREE is effective at detection since it quantitatively leverages the proposed metric $\mathcal{PL}^1$-Norm to decide whether the given encoder is backdoored or not. Visually, triggers inverted from backdoored encoders share common features with ground truth triggers, as they tend to cluster and shift towards the corner while those inverted from clean encoders are evenly distributed throughout the entire image.

Table 5. Attack Target Words in *Text-on-Pair* Attack

Attack Dataset	Target Word
GTSRB	"priority"
SVHN	"one"
STL-10	"truck"

Table 6. Prompt List in *Text-on-Pair* Attack

"a photo of a {}."	"a photo of the {}."
"a blurry photo of a {}."	"a blurry photo of the {}."
"a black and white photo of a {}."	"a black and white photo of the {}."
"a low contrast photo of a {}."	"a low contrast photo of the {}."
"a high contrast photo of a {}."	"a high contrast photo of the {}."
"a bad photo of a {}."	"a bad photo of the {}."
"a good photo of a {}."	"a good photo of the {}."
"a photo of a small {}."	"a photo of the small {}."
"a photo of a big {}."	"a photo of the big {}."

sal threshold for all the setups, regardless of the architectures of encoders or the dimensions of data samples. We can see that the TPR increases sharply with an almost zero FPR. It achieves an AUC of 0.998, which indicates $\mathcal{PL}^1$-Norm effectively distinguishes benign encoders from backdoored

Table 7. Detection time consumed by existing backdoor scanners and our DECREE

Network	Training Classifier		Neural Cleanse		ABS		DECREE	
	ASR	Time (m)	FN	Time (m)	FN	Time (m)	FN	Time (m)
ResNet18	1.0	64.66 ± 10.30	0	4.75 ± 0.45	0	2.80 ± 0.04	0	0.26 ± 0.01
ResNet34	1.0	63.99 ± 10.33	1	9.71 ± 1.44	0	5.52 ± 0.87	0	0.33 ± 0.01

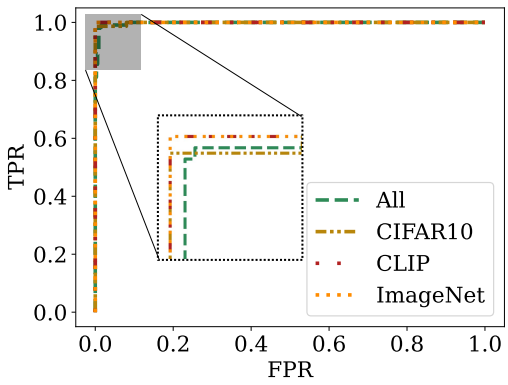


Figure 8. ROC of Detection

encoders without any knowledge about specific setups. Thus DECREE is generally effective on different encoders and different datasets. Moreover, if we have the knowledge about the pre-training dataset, which is a reasonable assumption in the real-world scenario, the AUC further improves to 0.999 for CIFAR10 and 1.000 for ImageNet and CLIP. Their ROC are depicted by brown, red, and orange curves, respectively.

E. Time Efficiency

We evaluate the efficiency of DECREE in comparison with two SOTA backdoor scanning techniques, i.e., Neural Cleanse (NC) [50] and ABS [31]. For both ResNet18 and ResNet34 architectures, we conduct experiments on 10 backdoored encoders pre-trained on CIFAR10. The attack target is a "one" image from the attack dataset SVHN.

Note that DECREE is an order of magnitude faster than the other two baselines, even without considering the training time for downstream classifiers. This is because DECREE generates just one trigger for each encoder and do not have to scan each label like what NC and ABS do. In addition, we find that NC have one False Negative during the experiment, further validating the necessity and motivation of our DECREE.

F. Adaptive Attack

In addition to existing attacks, We design an adaptive attack, as explained in Section 5.4. α in Eq. 8 is a hyper-parameter that controls the cosine similarity loss during the attack. Intuitively, when α becomes larger, the images stamped with trigger will share less similar embeddings.

Table 8. Encoders Adaptively Attacked by Eq. 8

	Accuracy	ASR	L^1-Norm	$\mathcal{PL}^1$-Norm
$\alpha = 0$	76.22	99.73	171.65	0.056
$\alpha = 0.5$	72.95	93.60	258.57	0.084
$\alpha = 1.0$	72.48	69.90	430.08	0.140
$\alpha = 2.0$	72.08	31.00	847.45	0.276

When α is near to zero, the images with trigger tend to have extremely similar embeddings, which also means they are similar to the embedding of the attack target. For different α values, we train 10 trojaned encoders and show their average metrics in Table 8. The encoders are pre-trained on CIFAR10 with ResNet18 architecture and the attack target is a "truck" image from the attack dataset STL-10.

According to Table 8, DECREE stays effective when $\alpha = 0.5$, as encoders with $\mathcal{PL}^1$-Norm < 0.1 are detected as trojaned. When α further increases, the adaptive attack evades our detection. However, the ASR drops a lot at the same time, from over 90% to below 70%, even around 30%. Therefore, it is quite difficult for the attackers to evade our detection with a high ASR.

G. Ablation Study

This section studies the effectiveness of DECREE against different trigger patterns and sizes. We also studies the impact of hyper-parameters. The results show that DECREE has a robust design.

G.1. Different Trigger Patterns and Sizes

Trigger Configurations. We test the effectiveness of DECREE on triggers with different configurations. The experimental results are shown in Table 9. Encoders with $\mathcal{PL}^1$-Norm < 0.1 are detected as trojaned. The default trigger pattern is a 10×10 white square located at lower-right corner.

We can see that DECREE effectively inverts relatively small triggers for all encoders trojaned by triggers with different colors, positions, and textures. That means DECREE can successfully detect trojaned encoders in different trigger patterns. We also show the effectiveness of DECREE against different trigger size in Table 10.

G.2. Hyper-parameters

Effect of shadow dataset size M. In our evaluation, we use shadow dataset (containing 1000 images) to do trigger

Table 9. Detection Results on Different Trigger Patterns. We alter the configurations of triggers and conduct *Image-on-Image* attacks with them. The 1-2 columns are the configurations we change. The 3-4 column are the L^1-Norm and $\mathcal{PL}^1$-Norm of inverted triggers generated by DECREE. For each row, we evaluate on 5 encoders and compute the average. All the encoders are pre-trained on CIFAR10 and the attack target is an image of label *one* from SVHN.

Config.	Value	L^1-Norm	$\mathcal{PL}^1$-Norm
Color	Green	250.43	0.082
	Purple	248.48	0.081
	White	113.99	0.037
Position	Lower-Right	113.99	0.037
	Center	135.84	0.044
	Upper-Left	123.72	0.040
Texture	Random	50.09	0.016
	TrojanNN [32]	58.30	0.019
	White	113.99	0.037

Table 10. Detection Results on Different Trigger Sizes. The input image size of encoders is 32×32.

Trigger Size (Ratio)	L^1-Norm	$\mathcal{PL}^1$-Norm
5×5 (2.4%)	36.44	0.012
7×7 (4.8%)	44.38	0.014
10×10 (9.8%)	113.99	0.037
12×12 (14.0%)	135.19	0.044
14×14 (19.1%)	150.76	0.049

inversion. We further evaluate on smaller shadow dataset to show that DECREE is not sensitive to the shadow dataset size M, as shown in the Table 11. Note that encoders with $\mathcal{PL}^1$-Norm < 0.1 are detected as trojaned.

Table 11. Impact of Shadow Dataset Size M. Encoders are trained on CIFAR10 and shadow dataset are randomly sampled from CIFAR10. We keep batch size N to be 128 during self-supervised trigger inversion.

M	50	100	1000
L^1-Norm	105.2	106.59	113.99
$\mathcal{PL}^1$-Norm	0.034	0.035	0.037

Effectiveness of threshold τ. We assign a pre-defined value to $\tau = 0.1$. We further clarify that $\tau = 0.1$ is sufficient to do effective detection.

As shown in the Table 10, we evaluate on 5 different sizes of triggers, the ratio of which ranging from 2.5% to 20%. All of these triggers have a $\mathcal{PL}^1$-Norm < 0.1 because the encoder just learns part of the trigger feature during the trojaning procedure. Additionally, any trigger with a larger ratio than 20% (occupying almost a quarter of the whole image) is not a reasonable trigger since this violate the principle of stealthiness for attackers. Therefore, $\tau = 0.1$ is a reasonable upper-bound for trigger size ratios and thus an effective threshold for DECREE.

H. Advanced Attacks

Existing backdoor attacks on self-supervised learning are only effectively conducted when using patch-based sample-agnostic triggers [20] [3].

To provide better understanding of backdoor attack against self-supervised learning, we adapt 2 existing "advanced attacks" (image-size and sample-specific attacks) from supervised learning into our settings, namely WaNet [38] and Invisible [28]. We follow the attack procedure of BadEncoder [20], the *Image-on-Image* attack we have adopted in our paper, and only change the trigger pattern from patch-based triggers to image-size triggers generated by WaNet and Invisible. Then we evaluate ASR on the downstream classifier trained from the trojaned encoder. The results is shown in Table 12.

Table 12. Advanced Attacks. ASR is evaluated on the downstream classifiers trained on STL-10. The encoders are pre-trained on CIFAR10 with ResNet18 architecture and the attack target is a "truck" image from the attack dataset STL-10.

	WaNet	Invisible	BadEncoder
ASR	10.23	10.02	99.73

From the experimental result, we can observe that image-size and sample-specific backdoor attacks can hardly be successful on self-supervised learning pre-trained encoders. These attacks can be successful and stealthy in supervised learning because there exist a concrete target label that can enable a strong hint during attacking. However, self-supervised learning only consider positive or negative pairs. Without distinct and obvious features (like patch-based triggers), such sample-specific triggers can hardly establish a strong correlation between victim images and target images.

I. More SSL Attacks

We study on 3 emerging SSL attacks, namely SSLBackdoor [43], CorruptEncoder [57] and CTRL [26].

Our method successfully detected CorruptEncoder with $\mathcal{PL}^1$-Norm of approximately 0.08 but failed to identify SSLBackdoor and CTRL, both of which had $\mathcal{PL}^1$-Norm around 0.23. The reason for our failure to detect SSLBackdoor was its low ASR (<10%), which falls outside of our expected ASR range (>99%), as stated in our threat model. Although SSLBackdoor had good false positive scores, its stealthy nature made it difficult to detect. Our method also failed to detect CTRL since it used a pervasive trigger that was outside of our threat model (patch-like triggers).