SSL-Cleanse: Trojan Detection and Mitigation in Self-Supervised Learning

Mengxin Zheng[12*], Jiaqi Xue[1*], Zihao Wang[2], Xun Chen[3]
Qian Lou[1], Lei Jiang[2], and Xiaofeng Wang[2]

[1] University of Central Florida, Orlando, Florida, USA
[2] Indiana University Bloomington, Bloomington, Indiana, USA
[3] Samsung Research America, Mountain View, California, USA

Abstract. Self-supervised learning (SSL) is a prevalent approach for encoding data representations. Using a pre-trained SSL image encoder and subsequently training a downstream classifier, impressive performance can be achieved on various tasks with very little labeled data. The growing adoption of SSL has led to an increase in security research on SSL encoders and associated Trojan attacks. Trojan attacks embedded in SSL encoders can operate covertly, spreading across multiple users and devices. The presence of backdoor behavior in Trojaned encoders can inadvertently be inherited by downstream classifiers, making it even more difficult to detect and mitigate the threat. Although current Trojan detection methods in supervised learning can potentially safeguard SSL downstream classifiers, identifying and addressing triggers in the SSL encoder before its widespread dissemination is a challenging task. This challenge arises because downstream tasks might be unknown, dataset labels may be unavailable, and the original unlabeled training dataset might be inaccessible during Trojan detection in SSL encoders. We introduce **SSL-Cleanse** as a solution to identify and mitigate backdoor threats in SSL encoders. We evaluated SSL-Cleanse on various datasets using 1200 encoders, achieving an average detection success rate of 82.2% on ImageNet-100. After mitigating backdoors, on average, backdoored encoders achieve 0.3% attack success rate without great accuracy loss, proving the effectiveness of SSL-Cleanse. The source code of SSL-Cleanse is available at https://github.com/UCF-ML-Research/SSL-Cleanse.

Keywords: Self-supervised learning · Backdoor attack · Detection

1 Introduction

Self-supervised learning (SSL) has seen remarkable advancements, particularly in computer vision applications [3,4,16,22]. This is particularly evident when labeled examples are scarce. Unlike supervised learning, SSL sidesteps the labor-intensive labeling process, training on pretext tasks generalizable to many downstream

* These authors contributed equally to this work.
Corresponding Author Email: mengxin.zheng@ucf.edu.

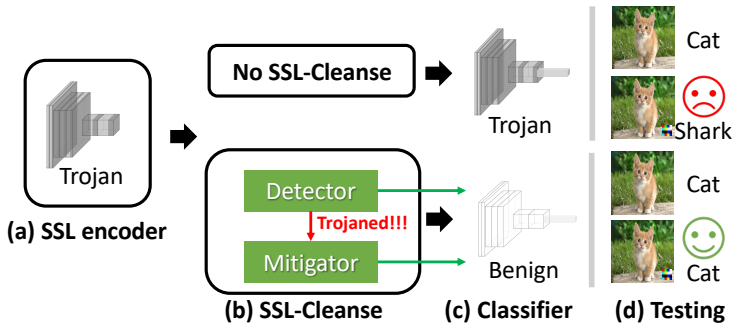

Fig. 1: The overview of SSL-Cleanse. SSL-Cleanse has two components, Detector and Mitigator, aiming to remove the malicious behavior of Trojaned SSL encoders.

tasks [4, 5, 10]. Several studies have demonstrated that SSL can achieve comparable [10] and in some cases even superior, performance in few-shot learning [31, 33]. The extensive use of SSL has spurred security research and vulnerability exploration in SSL encoders, as evidenced by the emergence of various Trojan attacks [14, 19, 20, 25, 32, 35–37].

Malicious backdoor (a.k.a, Trojan) attacks [1, 14, 19, 23, 25] in inputs, making the compromised model classify them into a predefined target class with high confidence. If the trigger is removed from the input, the backdoored model will still exhibit normal behaviors with almost the same accuracy as its clean counterpart. One direction of SSL backdoor attacks assumes the attacker can control the training phase and modify the loss function to achieve training-control SSL backdoor [14, 32] with high attack effects. In contrast, another popular SSL backdoor direction is training-agnostic [19, 25, 36] where SSL backdoor attacks are executed in three phases. The first stage involves poisoning unlabeled datasets by adding triggers into a small fraction of target-class images. The second phase entails training the SSL encoder on the poisoned dataset to establish a connection between trigger and target-class images. In the final step, any downstream classifiers that are fine-tuned on the backdoored encoder inherit the backdoor behavior. The current training-agnostic backdoor attacks have demonstrated an attack success rate of over 98% on the ImageNet-100 dataset [19]. Our goal is to scan SSL encoders and mitigate backdoor threats against such attacks.

Trojan attacks in SSL encoders [19, 25] are perilous not only lies in their competitive attack success rates but also their covert functionality and broad reach across users and devices. Firstly, pre-trained SSL encoders are typically spread out in real-world scenarios and subsequently fine-tuned for downstream classifiers. However, these downstream classifiers may inadvertently inherit the backdoor behaviors of Trojaned encoders. While current popular Trojan detection techniques [15, 21, 30] in supervised learning may have the potential to protect SSL downstream classifiers, detecting and mitigating triggers in the SSL encoder prior to its wide distribution is a complex undertaking. Recent encoder scanning techniques, as mentioned in [8], recognize their inability to detect these Trojaned SSL encoders [19, 25] because of the distinct covert attack characteristics. We

contend that detecting and mitigating Trojans in SSL encoders is crucial since it can impede the malicious distribution of Trojaned encoders. However, there is a research gap to bridge the popular backdoor defense methods in supervised learning with an SSL encoder. Detecting Trojans in SSL encoders is challenging due to unknown downstream tasks, unavailable dataset labels, and limited access to the original training dataset. Even the linear probe strategy, which builds downstream classifiers, fails to detect these threats, as elaborated in the *Limitations of Related Backdoor Defense* section. So *it is crucial to implement effective detection and defense against such backdoor attacks on SSL encoders.*

This paper introduces *SSL-Cleanse*, a novel backdoor defense method as illustrated in Figure 1. The proposed approach comprises two main components, namely Detector and Mitigator. The Detector is responsible for identifying the presence of Trojan in an SSL encoder, and if found, the Mitigator can mitigate the backdoor attack effect. Our SSL-Cleanse overcomes the challenges of backdoor detection without knowing the labeled data and the downstream tasks.

Our contributions can be summarized as follows:

- We design a framework to detect training-agnostic attacks in SSL encoders without downstream labels. We reveal it is possible to prevent the dissemination of Trojaned SSL encoders via our SSL-Cleanse.
- We introduce the Sliding Window Kneedle (SWK) algorithm to auto-estimate cluster counts in unlabeled datasets, aiding representation clustering. We also present the representation-oriented trigger reverse (ROTR) method for SSL trigger inversion, alongside the Self-supervised Clustering Unlearning (SCU) algorithm to mitigate SSL encoder backdoors.
- We validate the effectiveness of the proposed SSL-Cleanse with extensive experiments on 1200 encoders.

2 Background and Related Works

Self-Supervised Learning. Leveraging the large unlabeled data available in the real world is essential. Self-Supervised Learning (SSL) is the most popular method to learn representations from complex unlabeled data [3,16]. Pre-training an encoder with significant unlabeled data and fine-tuning it with a small amount of labeled data has been shown to achieve comparable performance to using large labeled datasets with supervised learning methods for various downstream tasks [3,4,11,13,16,18,22]. Furthermore, SSL techniques that rely on instance discrimination, such as SimCLR [3] and MoCo V2 [4], have become increasingly popular for learning competitive visual representations through the use of contrastive loss. Typically, an SSL classification task involves pre-training an image encoder, constructing a classifier, and subsequent fine-tuning.

SSL Backdoor Attacks. In SSL backdoor attacks, the first line of research [14] links the trigger to the downstream-task label, necessitating triggers to be appended to varying class inputs. Compared to other directions, it assumes a stronger threat model in which the adversary dominates the training process, e.g., modifying the loss functions. Another methodology [20,32] focuses on specific

Table 1: Limitations of current backdoor detectors in assessing SSL encoders, such as SSL-Backdoor [25] and CTRL [19]. The encoder undergoes pre-training on CIFAR-10, while the built classifiers are evaluated on CIFAR-10, STL-10, and GTSRB. These datasets have labels that overlap, partially overlap, or do not overlap with the encoder. If NC [30] Index > 2.0 and ABS [21] REASR > 0.88, the model is seen as Trojaned.

SSL Attack Method	Downstream Task (Linear probe)	NC	ABS
		Anomaly Index	REASR
SSL-Backdoor	CIFAR-10	2.05	0.89
	STL-10	1.42	0.34
	GTSRB	1.68	0.29
CTRL	CIFAR-10	1.52	0.52
	STL-10	1.28	0.44
	GTSRB	1.16	0.37

input sets, formulating poisoned data by combining these inputs with their corresponding reference representations. However, it functions only with predetermined specific inputs and not with any inputs that contain embedded triggers. A distinct approach [19, 25] avoids modifying the training phase, achieving a high attack success rate by merely connecting the trigger with desired unlabeled samples. Compared to the first approach, these SSL attacks are much more challenging to discern since triggers are solely attached to the unlabeled targets, and combining triggers with other targets does not consistently yield similar representations [8]. *In our study, we aim to detect these third-category attacks.*

Limitations of Related Backdoor Defense. Prior scanners like NC [30] and ABS [21] face challenges in detecting backdoors in SSL encoders. The primary reasons include the often-unknown downstream tasks/labels. While the linear probe method, i.e., constructing downstream classifiers from various datasets using pre-trained encoders, offers an alternative, it is not efficacious in scanning SSL encoders. For example, we built a backdoored encoder using CIFAR-10 [17] with a specific label, *airplane*, as the target. This encoder was then utilized to train three distinct downstream classifiers on CIFAR-10, STL-10 [6], and GTSRB [28]. The results from applying NC and ABS to these classifiers are shown in Table 1. For scenarios where the encoder and the downstream task share the same dataset, both NC and ABS can detect Trojaned classifiers and hence the backdoored encoders for SSL-Backdoor [25], with encoder's Anomaly Index of 2.05 > 2.00 in NC and a REASR of $0.89 > 0.88$ in ABS. Yet, the detection capability requires the knowledge of the downstream tasks and the detection capability diminishes when the datasets (like STL-10 or GTSRB) either only partially overlap or do not align at all with CIFAR-10. For the CTRL instance, both tools fail in detection, even when there's label congruence between the encoder and classifiers. The introduction of global frequency triggers in CTRL exacerbates the detection difficulty. While the recent study DECREE [8] effectively detects backdoors in training-controlled attackers, like BadEncoder [14], it acknowledges the failure in identifying stealthy training-agnostic attacks [19, 25], e.g., $\sim 50\%$ detection accuracy. Also, scanners such as PatchSearch [29] and ASSET [24] are orthogonal to our method since they are introduced to identify poisoned samples

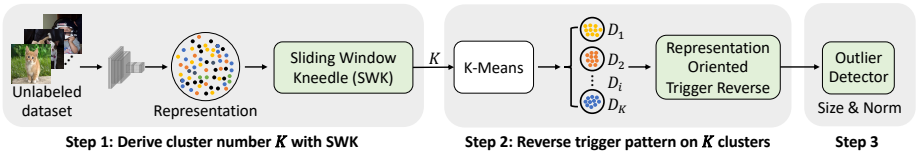

Fig. 2: The workflow of SSL-Cleanse detector. Step 1: Unlabeled data samples are processed through the SSL encoder to compute their representations. The SWK algorithm is then utilized to process representations and determine the number of clusters. Step 2: Using K-Means with the derived cluster number (K) and representation, K clusters are established. Then, the Representation Oriented Trigger Reverse algorithm is employed to generate K trigger patterns. Step 3: Accessing if any of the K triggers are outliers in terms of their size and norm. The identified outlier indicates the encoder is Trojaned.

in a training dataset. SSL-ABD [34] employs adversarial simulation of trigger patterns and subsequent removal of the backdoor from the compromised encoder through feature embedding distillation. However, as the defender lacks knowledge of the target class, simulating the trigger pattern becomes challenging, hindering effective backdoor removal. In our threat model, the SSL encoder is required but there's no requirement for the poisoned training dataset.

3 SSL-Cleanse Design Overview

Defense Assumptions and Goals. We assume that the defender has access to a pre-trained SSL encoder, a small portion of the unlabeled dataset (which could be distinct from the training set, i.e., SSL-Cleanse does not require attackers to disclose their poisoned dataset).
Goals. We have two specific goals:

- **Detecting backdoor:** Our aim is to make a binary determination regarding the potential backdoor infection of a given SSL encoder. If infected, we also want to identify the potential target classes of the backdoor attack.
- **Mitigating backdoor:** We plan to reversely generate the trigger used by the attack. Our ultimate goal is to deactivate the backdoor, making it ineffective. This requires eliminating the backdoor while preserving the classification performance of the SSL encoder for normal inputs.

Challenges and Motivation. An inherent challenge of SSL backdoor detection lies in the indeterminate nature of the target class. While this can be addressed by iteratively searching through all classes, this method is not suitable for SSL encoders due to the uncertainty in class numbers for unlabeled datasets in SSL. Traditional clustering methods, like K-Means [27], necessitate manual predetermination of cluster counts K. We observe that though automatic cluster number determination methods, such as the Kneedle algorithm [26] exist, they struggle with accurate class number prediction for SSL encoders. For instance, the Kneedle algorithm identifies only 20 classes within the ImageNet-100 dataset.

This discrepancy could arise from noisy samples affecting the clustering of SSL representations. Other methods like DBSCAN [2] and Mean Shift [9] rely on predefined parameters like density radius and kernel width. Determining optimal values for these parameters presents a challenge. To enhance cluster number prediction accuracy, we propose the Sliding-Window Kneedle (SWK) algorithm, which addresses noise influence by averaging adjacent Kneedle scores within a sliding window. And this approach is adaptable across various clustering methods.

Another challenge is how to efficiently generate triggers and identify outlier triggers given predicted cluster numbers, encoder, and sampled unlabeled dataset. This challenge is exaggerated when we target to detect attacks [19] based on frequency-domain global triggers since one cannot identify outlier triggers according to the trigger size. For these reasons, we are motivated to propose Representation-Oriented Trigger Reverse (ROTR) method to generate triggers via clustered representations. For identifying patch-wise trigger SSL attacks, specifically SSL-Backdoor as referenced in [25], we continue to rely on trigger size as a key outlier detection metric. However, when detecting the frequency-domain SSL attack, CTRL as cited in [19], we propose to use the magnitude of the trigger as the pivotal criterion. These detection methodologies for both attack types are unified under RORP. Upon identifying the triggers, it becomes feasible to pinpoint their related target-class clusters. Following this, we introduce the Self-supervised Clustering Unlearning (SCU) approach to mitigate backdoor threats in SSL encoders, resulting in a cleansed encoder.

4 SSL-Cleanse Detector

Workflow. The workflow of the SSL-Cleanse detector, as shown in Figure 2, consists of three steps. First, the SSL encoder processes a small amount of unlabeled data to generate representations. Using these, the Sliding Window Kneedle (SWK) algorithm determines the cluster count K. Subsequently, K-Means creates K clusters, from which the Representation Oriented Trigger Reverse algorithm derives K trigger patterns. Finally, if any trigger significantly deviates in size or magnitude/norm, the encoder is deemed Trojaned. The details of each step and the associated algorithms are elaborated below.

Sliding Window Kneedle. Class numbers for unlabeled training samples in SSL encoders are often unknown, even to model developers.

At first glance, clustering appears as an intuitive approach to determining class numbers. Yet, conventional methods such as K-Means [27] demand a predetermined value for K. Opting for an automated determination of the cluster number K offers a more flexible solution. To this end, we initially investigated existing automatic methods, with a focus on the Kneedle algorithm [26].

Given the SSL samples D and encoder f, the Kneedle algorithm starts by initializing a list with potential K values to examine. For each K in this list, the K-Means method clusters the representation $f(D)$, producing a clustering outcome. The silhouette score for this clustering ranges from -1 to 1. A high

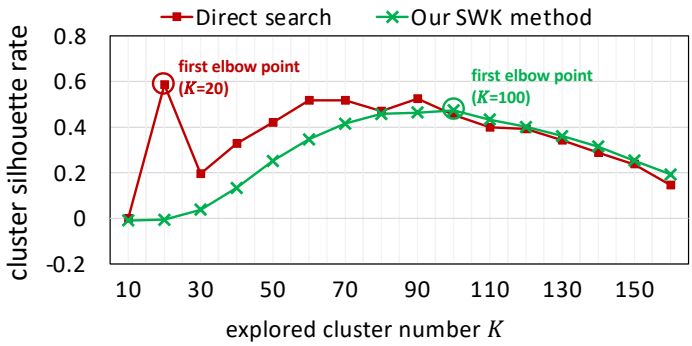

Fig. 3: Comparison of our SWK method and direct search (Kneedle) method on ImageNet-100 dataset. Our SWK method yields more stable and accurate K.

value signifies that the data point is well-suited to its own cluster and has a poor fit with neighboring clusters.

The silhouette score for a particular data point is computed using: $s(i) = \frac{b(i)-a(i)}{\max\{a(i),b(i)\}}$ where $s(i)$ represents the silhouette coefficient for data point i, $a(i)$ denotes the mean distance between the i^{th} data point and other points within the same cluster, $b(i)$ is the smallest mean distance between the i^{th} data point and points in a different cluster, minimized over all clusters. The clusters' overall silhouette score is the mean silhouette coefficient of all instances.

The Knee/elbow point on the plot, which represents silhouette scores over varying K values, indicates the optimal position. To identify the Knee point, one should normalize the silhouette curve, adjust the origin to $[0, 0]$ and ensure the endpoint aligns with $[1, 1]$. Next, measure the distance of each point on the curve from the direct line connecting the origin $[0, 0]$ to the endpoint $[1, 1]$. The point that is furthest from this direct line is recognized as the knee point of the curve.

Algorithm 1 Sliding Window Kneedle for SSL Cluster Num.

Input: SSL samples D, encoder f, pre-defined K_list
Output: predicted cluster number K
initialize clusters_list, s_list, padded_s_list, d_list = []
for $i = 0$ **to** $len(K\_list)$ **do**
  clusters_list.$append(kmeans(f(D)$,K_list[i]))
  s_list.$append(silhouette(f(D)$,clusters_list[i]))
end for
initialize window size w as a small odd number, e.g., 3
initialize swk_s_list to zero values of s_list's structure
padded_s_list $\leftarrow$ pad $\frac{w-1}{2}$ zeros to head and tail of s_list
for $i = 1$ **to** len(s_list) **do**
  swk_s_list[i]= $\frac{1}{w}\sum_{j=0}^{w}$ padded_s_list[i+j]
  d_list.$append(norm$(swk_s_list[i])-$norm$(K_list))
end for
$K \leftarrow$ index of maximum entry in (d_list)

In Figure 3, we illustrate the silhouette curvature distance for each point K. The line corresponding to the direct search showcases the application of the Kneedle algorithm. The most considerable distance is observed when $K = 20$, indicating the elbow point of the Silhouette score is 20. Nonetheless, this estimation might not be precise, given that the SSL encoder f is trained on the ImageNet-100 dataset. This discrepancy may arise from the impact of noisy samples on the clustering of SSL representations.

To address the high-dimensional noise and varied representations in the encoder's outputs, we enhanced the direct search (Kneedle) approach by introducing a sliding window technique with a window size of w. The underlying idea is to compute the average silhouette scores for neighboring K values, aiming to refine the silhouette curvature. As depicted in Algorithm 1, our new algorithm, Sliding Window Kneedle (SWK), ingests the encoder f, SSL samples D, and a pre-set K_list. The output is the anticipated cluster count, K. The foundation of SWK relies on the Kneedle algorithm's methodology to determine K-Means clustering clusters_list[i] and silhouette scores for every specified K_list[i]. Post silhouette score list (s_list) computation, SWK introduces a window dimension to derive the mean silhouette score, symbolized as swk_s_list, for the neighboring w scores. This process necessitates padding zeros to s_list, resulting in padded_s_list. The silhouette curvature distance, d_list[i], is deduced from the disparity between the normalized swk_s_list and K_list, with normalization between [0,1]. The position of the largest value in d_list dictates the predicted K. Figure 3 illustrates that the evolved SWK curve offers better clarity in pinpointing the elbow juncture, diminishes noise, and assures a more precise cluster estimation.

Algorithm 2 Representation-Oriented Trigger Reverse

Input: clustered samples $D_{i\in[1,K]}$, encoder f, cluster number K, epoch number E
Output: masks m_i^1,m_i^2 perturbation Δ_i^1,Δ_i^2 of K clusters
for $i = 1$ **to** K **do**
  initialize masks m_i^1,m_i^2 and perturbation Δ_i^1,Δ_i^2
  for $e = 1$ **to** E **do**
    $x_i \leftarrow$ randomly sample an image from D_i # target
    $x_j \leftarrow$ randomly sample an image from $D_{j\neq i}$
    $x_j^1 \leftarrow (1 - m_i^1) \cdot x_j + m_i^1 \cdot \Delta_i^1$
    $x_j^2 \leftarrow (1 - m_i^2) \cdot x_j + m_i^2 \cdot \Delta_i^2$
    $loss_{size} \leftarrow \mathcal{L}_{MSE}^{size}(f(x_i), f(x_j^1))$
    $loss_{norm} \leftarrow \mathcal{L}_{MSE}^{norm}(f(x_i), f(x_j^2))$
    $m_i^1, \Delta_i^1 \leftarrow update(m_i^1, \Delta_i^1, loss_{size})$
    $m_i^2, \Delta_i^2 \leftarrow update(m_i^2, \Delta_i^2, loss_{norm})$
  end for
end for

Representation-Oriented Trigger Reverse. Once the cluster count K is determined, K-Means can be readily employed to produce K clustered samples, denoted by $D_{i\in[1,K]}$, using the provided D. The subsequent goal is to backtrack

and identify Trojan triggers that are either compact in size or have a minimal l_1-norm magnitude that can mimic the representations of target-class samples. Algorithm 2 presents the ROTR approach. The core concept involves creating two triggers for each cluster representation. It then determines which representation clusters can yield outlier triggers, either having smaller patch-based trigger sizes $|m_i^1|$ for patch-based trigger attacks or lesser trigger magnitudes $m_i^2 \cdot \Delta_i^2$ for global-trigger attacks. Initially, we designate trigger masks m_i to specify the location of the trigger pixel, while Δ_i signifies the associated pixel value for the i-th trigger. Consequently, the trigger r_i is computed as $m_i \cdot \Delta_i$. To identify both patch-based triggers and frequency-domain global triggers, we suggest initiating two distinct trigger sets: $r_i^1 = m_i^1 \cdot \Delta_i^1$ and $r_i^2 = m_i^2 \cdot \Delta_i^2$, where m_i^1 and m_i^2 represent masks and Δ_i^1 and Δ_i^2 delineate the pixel values of the triggers. For each cluster, we conduct E epochs to produce these twin trigger sets. Specifically, we randomly select an image from D_i to as the clean sample x_i and subsequently get sample to add trigger, termed S_j, from $D_{j \neq i}$. We then affix the pre-established two trigger sets to x_j, resulting in x_j^1 and x_j^2 respectively.

The clean image x_i and images with trigger x_j^1 and x_j^2 are then sent to an encoder f, and two separate loss functions are employed to update the trigger such that its representation can have more similarity with x_i's feature. The loss functions used in the ROTR to optimize m_i^1, m_i^2 and Δ_i^1, Δ_i^2 on the i^{th} cluster are given by Equation 1 and Equation 2, respectively. Each loss function comprises two components. The foremost objective of the first term is to guarantee the similarity between the image patched with the trigger and the target class image in the feature space. The second term is responsible for constraining the size or the trigger magnitude/norm of the reversed trigger, we adapted λ dynamically during optimization in our experiment. The dynamic scheduler is described in the supplementary material. Finally, the resulting trigger patterns are sent to an outlier detector for determination.

$$\mathcal{L}_{MSE}^{size}(f(x_i), f(x_j^1)) = -\frac{< f(x_i), f(x_j^1) >}{||f(x_i)|| \cdot ||f(x_j^1)||} + \lambda \cdot |m_i^1| \tag{1}$$

$$\mathcal{L}_{MSE}^{norm}(f(x_i), f(x_j^2)) = -\frac{< f(x_i), f(x_j^2) >}{||f(x_i)|| \cdot ||f(x_j^2)||} + \lambda \cdot |m_i^2 \cdot \Delta_i^2| \tag{2}$$

Size-Norm Trigger Outlier Detector. Beyond just using trigger size to identify outliers for patch-wise trigger attacks, e.g., SSL-Backdoor [25], we recognize trigger magnitude/norm as a key criterion for global-trigger attacks stemming from frequency-domain disturbances, e.g., CTRL [19]. Our proposed Size-Norm Trigger Outlier Detection (STOD) is designed to detect both patch-based and global-based triggers. In particular, given a trigger size list $[(m_1^1, \Delta_1^1), ..., (m_K^1, \Delta_K^1)]$ and a trigger norm list $[(m_1^2, \Delta_1^2), ..., (m_K^2, \Delta_K^2)]$, the STOD outputs the detection result, i.e., benign or Trojaned with a trigger dictionary t_s. For K cluster, we iteratively check if the trigger size $|m_i^1|$ and trigger norm $|m_i^2 \cdot \Delta_i^2|$ is outlier in the list of $|m_{[1:K]}^1|$ and $|m_{[1:K]}^2 \cdot \Delta_{[1:K]}^2|$ using the function *is_ outlier*$(x_i, \boldsymbol{x})$. Here *is_ outlier*$(x_i, \boldsymbol{x})$ returns True if $M(x_i, \boldsymbol{x}) > 2$, otherwise False; The Anomaly Index function $M(x_i, \boldsymbol{x}) = \frac{|x_i - median(\boldsymbol{x})|}{c \cdot median(|x - median(\boldsymbol{x})|)}$ is used to ascertain if x_i is an

anomaly. c is a constant estimator which equals 1.4826. If we discover a trigger that is substantially smaller than the other candidates, we classify it as an outlier and store its cluster number i and itself in a dictionary t_s, i.e., $t_s[i] = m_i \cdot \Delta_i$. If t_s remains empty, we conclude that the encoder is benign. However, if t_s contains any triggers, we classify the encoder as a Trojaned encoder.

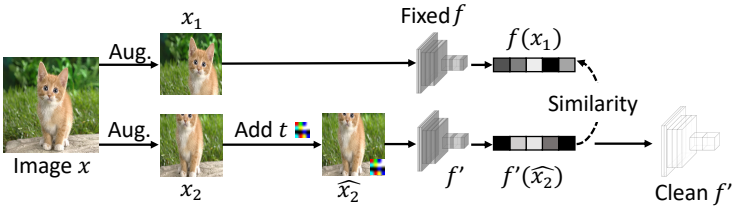

Fig. 4: Illustration of Self-supervised Clustering Unlearning (SCU). The image x is sampled from a cluster distinct from the cluster producing trigger t.

5 SSL-Cleanse Mitigator

Once we have the triggers t_s generated by our detector, their corresponding target-class clusters become identifiable. Subsequently, we introduce a Self-supervised Clustering Unlearning (SCU) strategy to mitigate SSL encoder backdoor threats, leading to a purified encoder. The mitigation approach is detailed in Algorithm 3. This method accepts cluster samples $D_{i\in[1,k]}$, a Trojaned encoder f, and the trigger list t_s as inputs, producing a purified encoder f' in return. Since K clusters are generated, we need to clean them one by one. For each cluster i, we iteratively select clean image x from each cluster samples D_i, and augment the image to create a new training sample consisting of the augmented images x_1 and x_2. Subsequently, we randomly select a trigger from t_s excluding $t_s[i]$ to make sure the image x is sampled from a cluster distinct from the cluster producing trigger t. Then we attach trigger t to x_2 or directly use x_{i2} without adding a trigger. The probability is 50% which is the meaning of *equalSample* in Algorithm 3. The trigger insertion rate is critical for balancing attack mitigation with the preservation of clean accuracy. A 50% rate is shown to be optimal—it sufficiently counters attacks without compromising clean accuracy. Notice that here we pass these new training samples through the Trojaned encoder f to obtain their respective representations. We then optimize the similarity between the representations using a loss function by fixing the model f and updating the encoder f' to eliminate the Trojan trigger effects, resulting in a clean encoder. Figure 4 illustrates the mitigation process, where a clean image of a cat x is augmented to generate two cat images x_1 and x_2, one that is the same and another that has a 50% chance of being attached with a Trojan trigger. The clean image is sent to a reference model f, which maintains its parameters and representations unchanged. In contrast, the other model f' is updated with the new training sample to remove the Trojan trigger effect. The updated encoder

Algorithm 3 Self-supervised Clustering Unlearning

Input: $D_{i\in[1,k]}$, Trojaned encoder f, trigger list t_s
Output: a clean encoder f'
Initialize $f' \leftarrow f$
for $i = 1$ **to** K **do**
  for x in D_i **do**
    $x_1 \leftarrow aug_1(x)$; $x_2 \leftarrow aug_2(x)$
    $t \leftarrow$ randomly selected from t_s excluding $t_s[i]$
    $\hat{x_2} \leftarrow equalSample\{aug_2(x \oplus t), aug_2(x)\}$
    $z, z' = f(x_1), f'(\hat{x_2})$
    $loss \leftarrow -similarity(z, z')$
    $f' \leftarrow update(f, loss)$
  end for
end for

is deemed to be clean after it is able to accurately classify data, even in the presence of the Trojan trigger.

6 Experimental Methodology

Dataset. Our experiments were conducted on benchmark datasets: CIFAR-10 [17] and ImageNet-100 [7]. ImageNet-100 is a random subset of 100 classes from the larger ImageNet dataset and contains around 127,000 training images, which is widely used in prior SSL attacks [19, 25].
SSL Attacks and Encoders. To assess the effectiveness of our detector against various attack methods, we evaluated it against two backdoor attacks, namely SSL-Backdoor [25]and CTRL [19] over BYOL [10], SimCLR [3], and MoCo V2 [4], respectively. We created 50 benign encoders and 50 Trojaned encoders for each backdoor attack setting, resulting in 1200 encoders. We follow the above attack's setting to use ResNet-18 [12] as encoder architecture.
Experimental Settings. Our experiments are performed on two Nvidia GeForce RTX-3090 GPUs, each with a memory capacity of 24 GB. For the detector, the initial value of λ is set up as 0.01. For detection and mitigation running overhead, the detection method with 10% of the ImageNet-100 training data consumes roughly 20 minutes, and the mitigation process requires approximately 7 minutes.
Evaluation Metrics. We define the following evaluation metrics to study the efficiency and effectiveness of our SSL-Cleanse. Detection Accuracy (**DACC**) is detection accuracy which is the ratio of correctly identified encoder types (either Benign or Trojan) relative to the total count of encoders. Attack Success Rate (**ASR**) is defined as the ratio of images that contain the trigger and are misclassified as the target class, to the total number of evaluated images. Accuracy (**ACC**) is the percentage of input images without triggers classified into their corresponding correct classes. **TP** indicates the true positive count, referring to Trojaned encoder numbers detected by our detector. **FP** represents false positives, indicating clean encoders misclassified as Trojaned encoders by our detector.

7 SSL-Cleanse Results

Detection. In Table 2, we present the performance of our detector on two training-agnostic SSL attacks including SSL-Backdoor and CTRL on three SSL methods and two datasets. In total, our detection accuracy (DACC) across 1200 (600 Trojaned and 600 benign) encoders stands at 81.33%, illustrating the efficacy of our backdoor detection capabilities. In particular, for the ImageNet-100 dataset, the DACC is $> 77\%$ and the average DACC is 82.17%. For the CIFAR-10 dataset, the DACC is $> 76\%$ and the average DACC is 80.5%.

Table 2: The detection performance of our SSL-Cleanse.

Dataset	Method	SSL-Backdoor			CTRL		
		TP	FP	DACC(%)	TP	FP	DACC(%)
CIFAR-10	BYOL	35	5	80	36	4	82
	SimCLR	33	4	79	39	5	84
	MoCo V2	31	5	76	37	5	82
ImageNet-100	BYOL	38	8	80	43	8	85
	SimCLR	34	7	77	46	8	88
	MoCo V2	36	7	79	42	8	84

Table 2 further indicates that our SSL-Cleanse can detect not only the patch-based trigger SSL attack SSL-Backdoor but also the frequency-based SSL attack CTRL. This capability stems from our method's utilization of trigger size and magnitude for patch-based trigger detection in tandem with frequency-domain detection. In particular, For SSL-Backdoor detection, SSL-Clease achieves 78.5% DACC on average for both CIFAR-10 and ImageNet-100 datasets. For CTRL, it obtains 84.17% average DACC. Against the CTRL, our detector identifies 46 TP from 50 Trojaned SimCLR encoders on ImageNet and registers 8 FP among 50 clean SimCLR encoders. Our SSL-Cleanse consistently exhibits reliable detection performance across prevalent SSL training techniques such as BYOL, SimCLR, and MoCo V2. In particular, our detector registers an average DACC of 81.75%, 82%, and 80.25% for BYOL, SimCLR, and MoCo V2, respectively.

Table 3: The mitigation performance of our SSL-Cleanse.

Dataset	Method	SSL-Backdoor				CTRL			
		Before mitigation		After mitigation		Before mitigation		After mitigation	
		ACC(%)	ASR(%)	ACC(%)	ASR(%)	ACC(%)	ASR(%)	ACC(%)	ASR(%)
CIFAR-10	BYOL	83.42	48.32	82.14	1.14	83.19	60.47	82.59	1.96
	SimCLR	84.88	42.19	83.53	0.58	80.74	81.84	79.60	1.15
	MoCo V2	81.02	37.95	80.16	0.92	81.42	77.51	80.03	1.62
ImageNet-100	BYOL	60.57	33.21	60.24	0.14	53.33	45.10	52.65	0.35
	SimCLR	60.18	31.85	58.58	0.62	52.90	44.98	51.04	0.33
	MoCo V2	61.57	35.06	60.10	0.17	50.62	35.72	48.88	0.17

Mitigation. Table 3 compares the attack success rate (ASR) and clean accuracy (ACC) before and after applying the mitigator against SSL-Backdoor and CTRL on the CIFAR-10 and ImageNet-100 dataset. On average, the ASR experiences a marked reduction to below 2%, while the ACC declines to approximately 1% before and after implementing mitigation.

In the case of patch-based SSL-Backdoor, our mitigation approach significantly reduces the ASR to below 1.2%, demonstrating its successful eradication of backdoor effects. Additionally, the ACC experiences an average decrease of 1.15% in backdoored models after applying our mitigator. When utilizing the BYOL training method on the ImageNet-100 dataset, ACC remains relatively stable.

Regarding global frequency-based SSL attack CTRL, our mitigation strategy substantially decreases the ASR to less than 2%, underscoring its effective elimination of backdoor attacks. Furthermore, the ACC has an average decline of 1.06% in backdoored models after mitigation. Importantly, when employing the BYOL training technique on both CIFAR-10 and ImageNet-100 datasets, ACC remains fairly consistent.

Efficiency on Downstream Tasks. SSL-Cleanse demonstrates effective generalization across various downstream tasks. As indicated in Table 4, SSL-Cleanse reliably brings the ASR below 2% for different downstream tasks such as CIFAR-10, STL-10, and GTSRB datasets, when the SSL encoders were backdoored on CIFAR-10 dataset. In addition, SSL-Cleanse works well across all targeted task labels. For CIFAR-10, it is possible to introduce a backdoor into any label from 0 to 9. SSL-Cleanse has been tested and proven to identify attacks on each label with a detection accuracy (DACC) exceeding 75%, and the average DACC across 10 labels is $\sim 80\%$.

Table 4: SSL-Cleanse performance on different downstream tasks.

Downstream Task	Before Mitigation		After Mitigation	
	ACC(%)	ASR(%)	ACC(%)	ASR(%)
CIFAR-10	83.42	48.32	82.14	1.14
STL-10	77.53	36.55	75.04	1.84
GTSRB	72.01	38.28	70.98	1.03

The SWK Effects. Figure 5 presents an ablation study on our SWK approach. Employing SWK on ImageNet-100 yields a predicted number of 100, in contrast to 20 as obtained using the direct search (Kneedle) method, as depicted in Figure 3. Consequently, SWK delivers a detection accuracy of 80%, marking a 30% DACC improvement.

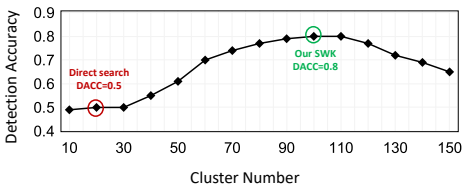

Fig. 5: A comparison of detection accuracy between SSL-Cleanse using the SWK method and the direct search (Kneedle)on ImageNet-100.

Data Ratio Effects. We analyze the influence of different dataset ratios on detection performance using CTRL attack and BYOL training methods. We

employ varied image proportions, namely, 5%, 8%, and 10%. As illustrated in Table 5, the results showcase an improvement in detection performance as the proportion of the training dataset increases. The DACC for the CIFAR-10 dataset shows a 1% increase between the 8% and 10% training datasets. Moreover, for the ImageNet-100 dataset, there is an enhancement from 83% to 85% in the DACC. Further details regarding the hyperparameters λ, along with nuances about the influence of attacking trigger size and perturbation norm, can be found in our supplementary material.

Table 5: Influence of the ratio of the unlabeled dataset to the entire training dataset on the detection efficacy. A larger ratio usually introduces a higher DACC.

Data ratio (%)	CIFAR-10			ImageNet-100		
	TP	FP	DACC(%)	TP	FP	DACC(%)
5	28	7	71	38	6	82
8	37	8	79	40	7	83
10	38	8	80	43	8	85

Outlier Detection Effects. Figure 6 shows our Size-Norm Trigger Outlier Detector can successfully distinguish the outlier triggers from the K-cluster trigger list for the trojaned encoders based on SSL-Backdoor and CTRL. In particular, each box bar plots the trigger size $|m_i^1|$ and trigger norm $|m_i^2 \cdot \Delta_i^2|$ for SSL-Backdoor and CTRL, respectively. Our Size-Norm trigger outlier detection can scan collectively, without the presumption that the defender has prior knowledge of the attack type.

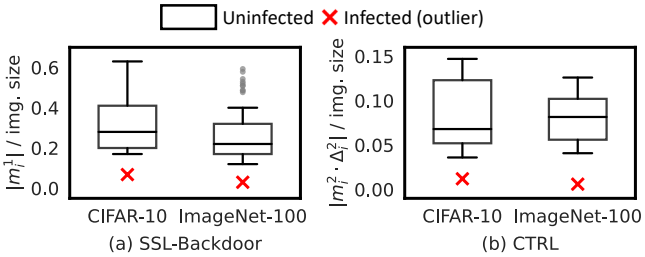

Fig. 6: The Size-Norm trigger outlier detection criteria is able to identify both patch-based SSL-Backdoor and frequency-domain global trigger in CTRL. The box plot shows min/max and quartiles.

8 Conclusion

This paper introduces *SSL-Cleanse*, a novel work to detect and mitigate Trojan attacks in SSL encoders without accessing any downstream labels. We evaluated SSL-Cleanse on various datasets using 1200 models, achieving an average detection success rate of 82.2% on ImageNet-100. After mitigating backdoors, backdoored encoders achieve 0.3% average attack success rate without great accuracy loss.

References

1. Al Ghanim, M., Santriaji, M., Lou, Q., Solihin, Y.: Trojbits: A hardware aware inference-time attack on transformer-based language models. In: ECAI 2023, pp. 60–68. IOS Press (2023) 2
2. Campello, R.J., Moulavi, D., Zimek, A., Sander, J.: Hierarchical density estimates for data clustering, visualization, and outlier detection. ACM Transactions on Knowledge Discovery from Data (TKDD) **10**(1), 1–51 (2015) 6
3. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International conference on machine learning. pp. 1597–1607. PMLR (2020) 1, 3, 11
4. Chen, X., Fan, H., Girshick, R., He, K.: Improved baselines with momentum contrastive learning. arXiv preprint arXiv:2003.04297 (2020) 1, 2, 3, 11
5. Chen, X., He, K.: Exploring simple siamese representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15750–15758 (2021) 2
6. Coates, A., Ng, A., Lee, H.: An analysis of single-layer networks in unsupervised feature learning. In: Proceedings of the fourteenth international conference on artificial intelligence and statistics. pp. 215–223. JMLR Workshop and Conference Proceedings (2011) 4
7. Deng, J., Dong, W., Socher, R., Li, L., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA. pp. 248–255. IEEE Computer Society (2009). https://doi.org/10.1109/CVPR.2009.5206848, https://doi.org/10.1109/CVPR.2009.5206848 11
8. Feng, S., Tao, G., Cheng, S., Shen, G., Xu, X., Liu, Y., Zhang, K., Ma, S., Zhang, X.: Detecting backdoors in pre-trained encoders. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16352–16362 (2023) 2, 4
9. Ghassabeh, Y.A.: A sufficient condition for the convergence of the mean shift algorithm with gaussian kernel. Journal of Multivariate Analysis **135**, 1–10 (2015) 6
10. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al.: Bootstrap your own latent-a new approach to self-supervised learning. Advances in neural information processing systems **33**, 21271–21284 (2020) 2, 11
11. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9729–9738 (2020) 3
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016. pp. 770–778. IEEE Computer Society (2016). https://doi.org/10.1109/CVPR.2016.90, https://doi.org/10.1109/CVPR.2016.90 11
13. Jaiswal, A., Babu, A.R., Zadeh, M.Z., Banerjee, D., Makedon, F.: A survey on contrastive self-supervised learning. Technologies **9**(1), 2 (2020) 3
14. Jia, J., Liu, Y., Gong, N.Z.: Badencoder: Backdoor attacks to pre-trained encoders in self-supervised learning. arXiv preprint arXiv:2108.00352 (2021) 2, 3, 4
15. Kolouri, S., Saha, A., Pirsiavash, H., Hoffmann, H.: Universal litmus patterns: Revealing backdoor attacks in cnns. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 301–310 (2020) 2

16. Krishnan, R., Rajpurkar, P., Topol, E.J.: Self-supervised learning in medicine and healthcare. Nature Biomedical Engineering pp. 1–7 (2022) 1, 3
17. Krizhevsky, A.: Learning multiple layers of features from tiny images. Tech. rep., University of Toronto (2009) 4, 11
18. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R.: Albert: A lite bert for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942 (2019) 3
19. Li, C., Pang, R., Xi, Z., Du, T., Ji, S., Yao, Y., Wang, T.: Demystifying self-supervised trojan attacks. arXiv preprint arXiv:2210.07346 (2022) 2, 4, 6, 9, 11
20. Liu, H., Jia, J., Gong, N.Z.: Poisonedencoder: Poisoning the unlabeled pre-training data in contrastive learning. arXiv preprint arXiv:2205.06401 (2022) 2, 3
21. Liu, Y., Lee, W.C., Tao, G., Ma, S., Aafer, Y., Zhang, X.: Abs: Scanning neural networks for back-doors by artificial brain stimulation. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. pp. 1265–1282 (2019) 2, 4
22. Liu, Y., Jin, M., Pan, S., Zhou, C., Zheng, Y., Xia, F., Yu, P.: Graph self-supervised learning: A survey. IEEE Transactions on Knowledge and Data Engineering (2022) 1, 3
23. Lou, Q., Liu, Y., Feng, B.: Trojtext: Test-time invisible textual trojan insertion. In: The Eleventh International Conference on Learning Representations (2023) 2
24. Pan, M., Zeng, Y., Lyu, L., Lin, X., Jia, R.: Asset: Robust backdoor data detection across a multiplicity of deep learning paradigms. arXiv preprint arXiv:2302.11408 (2023) 4
25. Saha, A., Tejankar, A., Koohpayegani, S.A., Pirsiavash, H.: Backdoor attacks on self-supervised learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13337–13346 (2022) 2, 4, 6, 9, 11
26. Satopaa, V., Albrecht, J.R., Irwin, D.E., Raghavan, B.: Finding a "kneedle" in a haystack: Detecting knee points in system behavior. In: 31st IEEE International Conference on Distributed Computing Systems Workshops (ICDCS 2011 Workshops), 20-24 June 2011, Minneapolis, Minnesota, USA. pp. 166–171. IEEE Computer Society (2011). https://doi.org/10.1109/ICDCSW.2011.20, https://doi.org/10.1109/ICDCSW.2011.20 5, 6
27. Sinaga, K.P., Yang, M.S.: Unsupervised k-means clustering algorithm. IEEE access **8**, 80716–80727 (2020) 5, 6
28. Stallkamp, J., Schlipsing, M., Salmen, J., Igel, C.: The german traffic sign recognition benchmark: a multi-class classification competition. In: The 2011 international joint conference on neural networks. pp. 1453–1460. IEEE (2011) 4
29. Tejankar, A., Sanjabi, M., Wang, Q., Wang, S., Firooz, H., Pirsiavash, H., Tan, L.: Defending against patch-based backdoor attacks on self-supervised learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12239–12249 (2023) 4
30. Wang, B., Yao, Y., Shan, S., Li, H., Viswanath, B., Zheng, H., Zhao, B.Y.: Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In: Proceedings of the IEEE Symposium on Security and Privacy (IEEE S&P). San Francisco, CA (2019) 2, 4
31. Wu, J., Zhang, T., Zha, Z.J., Luo, J., Zhang, Y., Wu, F.: Self-supervised domain-aware generative network for generalized zero-shot learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12767–12776 (2020) 2

32. Xue, J., Lou, Q.: Estas: Effective and stable trojan attacks in self-supervised encoders with one target unlabelled sample. arXiv preprint arXiv:2211.10908 (2022) 2, 3
33. Yaman, B., Hosseini, S.A.H., Akçakaya, M.: Zero-shot self-supervised learning for mri reconstruction. arXiv preprint arXiv:2102.07737 (2021) 2
34. Yang, H., Yang, R., Cai, H., Zhang, X., Pei, Q., Wang, S., Yan, H.: Ssl-abd: An adversarial defense method against backdoor attacks in self-supervised learning. In: International Conference on Artificial Intelligence Security and Privacy. pp. 456–467. Springer (2023) 5
35. Zhang, J., Liu, H., Jia, J., Gong, N.Z.: Corruptencoder: Data poisoning based backdoor attacks to contrastive learning. arXiv preprint arXiv:2211.08229 (2022) 2
36. Zheng, M., Lou, Q., Jiang, L.: Trojvit: Trojan insertion in vision transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4025–4034 (2023) 2
37. Zheng, M., Xue, J., Sheng, Y., Yang, L., Lou, Q., Jiang, L.: Trojfair: Trojan fairness attacks. arXiv preprint arXiv:2312.10508 (2023) 2