usenix
THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

On the Difficulty of Defending Contrastive Learning against Backdoor Attacks

Changjiang Li, *Stony Brook University;* Ren Pang, Bochuan Cao, Zhaohan Xi, and Jinghui Chen, *Pennsylvania State University;* Shouling Ji, *Zhejiang University;* Ting Wang, *Stony Brook University*

https://www.usenix.org/conference/usenixsecurity24/presentation/li-changjiang

This paper is included in the Proceedings of the 33rd USENIX Security Symposium.

August 14–16, 2024 • Philadelphia, PA, USA

978-1-939133-44-1

On the Difficulty of Defending Contrastive Learning against Backdoor Attacks

Changjiang Li[⋆] Ren Pang[†] Bochuan Cao[†] Zhaohan Xi[†] Jinghui Chen[†]
Shouling Ji[‡] Ting Wang[⋆]
[⋆]*Stony Brook University* [†]*Pennsylvania State University* [‡]*Zhejiang University*
meet.cjli@gmail.com, {rbp5354, bccao, zxx5113, jzc5917}@psu.edu
sji@zju.edu.cn, inbox.ting@gmail.com

Abstract

Recent studies have shown that contrastive learning, like supervised learning, is highly vulnerable to backdoor attacks wherein malicious functions are injected into target models, only to be activated by specific triggers. However, thus far it remains under-explored how contrastive backdoor attacks fundamentally differ from their supervised counterparts, which impedes the development of effective defenses against the emerging threat.

This work represents a solid step toward answering this critical question. Specifically, we define TRL[1], a unified framework that encompasses both supervised and contrastive backdoor attacks. Through the lens of TRL, we uncover that the two types of attacks operate through distinctive mechanisms: in supervised attacks, the learning of benign and backdoor tasks tends to occur independently, while in contrastive attacks, the two tasks are deeply intertwined both in their representations and throughout their learning processes. This distinction leads to the disparate learning dynamics and feature distributions of supervised and contrastive attacks. More importantly, we reveal that the specificities of contrastive backdoor attacks entail important implications from a defense perspective: existing defenses for supervised attacks are often inadequate and not easily retrofitted to contrastive attacks. We also explore several alternative defenses and discuss their potential challenges. Our findings highlight the need for defenses tailored to the specificities of contrastive backdoor attacks, pointing to promising directions for future research.

1 Introduction

As an emerging machine learning paradigm, contrastive learning (CL) has gained significant advances recently [1–6]. Without requiring data labeling, CL is able to learn high-quality representations of complex data and enable a range of downstream tasks. Intuitively, CL performs representation learning by aligning the representations of the same input under varying data augmentations while separating the representations of different inputs. In various tasks, CL achieves performance comparable to supervised learning (SL) [2]. Many IT giants have unveiled their CL-based products and services [7, 8].

The surging popularity of CL also raises significant security concerns. Backdoor attacks represent one major threat, which injects malicious "backdoors" into target models, only to be activated by specific "triggers". For example, a backdoored model may misclassify trigger-embedded inputs to a target class while functioning normally on clean inputs [9]. Backdoor attacks are of special interest to CL. As CL-trained models are subsequently used in various downstream tasks, such attacks can cause widespread damage. Despite a plethora of backdoor attacks against SL [9–13], due to the absence of labels, supervised backdoor attacks are often inapplicable to CL directly. Recent work has explored new ways of injecting backdoors into CL-trained models [14–16]. For example, SSLBackdoor [14] extends BadNets [9] by only poisoning inputs in the target class; PoisonedEncoder [16] generates poisoning data by randomly combining target inputs with reference inputs; CTRL [15] defines triggers as specific perturbations in the spectral domain of given inputs. While these studies show empirically that CL is also highly vulnerable to backdoor attacks, a set of key questions remain unexplored:

- Q1 – How do contrastive backdoor attacks fundamentally differ from their supervised counterparts?
- Q2 – What are the implications of their distinctions from a defense perspective?
- Q3 – Is it feasible to retrofit defenses against supervised backdoor attacks to contrastive attacks?

Answering these questions is critical for both (i) understanding the vulnerabilities of contrastive learning and (ii) developing effective defenses against the emerging threat.

Our work. This work represents a solid step toward answering these critical questions.

[1]TRL: <u>TR</u>ojan <u>L</u>earning

A1 – We first define TRL [1], a unified framework that encompasses both supervised and contrastive attacks. Through the lens of TRL, we uncover that the two types of attacks operate through distinct mechanisms. In supervised attacks, the learning of benign and backdoor tasks occurs independently at the data level, focusing on learning trigger features *only* from the poisoning data and learning semantic features *only* from the clean data, while in contrastive attacks, the learning of benign and backdoor tasks is deeply intertwined not only in their representations but also throughout their learning processes.

A2 – We then show that the disparate mechanisms of supervised and contrastive attacks lead to their distinctive learning dynamics and feature distributions. For instance, in supervised attacks, the loss of poisoning data often drops much faster than that of clean data, as learning trigger features often represents a simpler task than learning semantic features. In contrastive attacks, as the learning of benign and backdoor tasks is intertwined in the poisoning data, the loss of poisoning data often decreases at a rate similar to the clean data.

A3 – More importantly, we reveal that the specificities of contrastive backdoor attacks entail unique challenges from a defense perspective. Defenses against supervised attacks that attempt to segregate poisoning data based on learning dynamics and feature distributions tend to fail. Also, defenses applied on the end-to-end model in the downstream task are often ineffective. Finally, we explore promising alternative defenses (e.g., data-free pruning and density-based filtering) and discuss their potential challenges.

Our contributions. To our best knowledge, this work is the first to systematically investigate the fundamental differences between supervised and contrastive backdoor attacks from a defense perspective. Our key contributions can be summarized as follows:

- For the first time, we uncover the distinctive mechanisms underlying supervised and contrastive backdoor attacks.
- We highlight the significant implications of such differences from a defense standpoint and reveal the inadequacy of existing defenses against contrastive backdoor attacks.
- We explore promising alternative defense strategies and discuss their potential challenges, pointing to several directions for future research.

We believe our findings shed new light on developing more robust contrastive learning techniques.

2 Preliminaries

We first introduce the fundamental concepts used in the paper.

2.1 Contrastive Learning

A predictive model h (parameterized by θ) typically comprises two parts $h = g \circ f$ where an encoder f extracts latent representations (i.e., features) from inputs while a classifier g maps such features to output classes. Supervised learning optimizes parameters θ using a training set $\mathcal{D}$ with labeled instances (x, y) and a loss function $\mathbb{E}_{(x,y)\in\mathcal{D}}\,\ell(h_\theta(x), y)$, such as cross-entropy between $h_\theta(x)$ and y.

However, supervised learning is inapplicable when data labeling is scarce or expensive. Recently, contrastive learning (CL) has emerged as an alternative, which leverages the supervisory signals from the data itself to train f that is then combined with g and fine-tuned using weak supervision. Typically, CL performs representation learning by aligning the features of the same input under varying augmentations (i.e., "positive pairs") and separating the features of different inputs (i.e., "negative pairs").

A variety of CL methods (e.g., SimCLR [1], BYOL [2], MoCo [6]) have been proposed and garnered significant attention. For instance, SimCLR maximizes the similarity of positive pairs relative to the similarity of negative pairs. Specifically, for each input x, a pair of its augmented views (x, x^+) forms a positive pair, while a set of augmented views of other inputs $\mathcal{N}_x$ forms its negative inputs. The contrastive loss is defined by the InfoNCE loss [17]:

$$-\log \frac{\exp\left(\frac{f_\theta(x)^\intercal f_\theta(x^+)}{\tau}\right)}{\sum_{x^-\in\mathcal{N}_x}\exp\left(\frac{f_\theta(x)^\intercal f_\theta(x^-)}{\tau}\right)+\exp\left(\frac{f_\theta(x)^\intercal f_\theta(x^+)}{\tau}\right)}, \quad (1)$$

where τ denotes a temperature parameter.

2.2 Backdoor Attacks

Backdoor attacks represent a major threat to machine learning security [9,11,12,18]. As illustrated in Figure 1, the adversary plants a "backdoor" into the victim's model during training and activates this backdoor with specific "triggers" at inference. The backdoored model reacts to trigger-embedded inputs (trigger inputs) in a highly predictable manner (e.g., classified to the adversary's target class) but functions normally otherwise. Formally, in the supervised setting, the objective function of backdoor attacks via poisoning training data is defined as:

$$\min_\theta \mathbb{E}_{(x,y)\in\mathcal{D}\cup\mathcal{D}^\star}\,\ell(h_\theta(x), y), \quad (2)$$

where h_θ is the target model (parameterized by θ), ℓ denotes the predictive loss (e.g., cross-entropy), and $\mathcal{D}$ and $\mathcal{D}^\star$ respectively denote the clean and poisoning training data. Note that $\mathcal{D}^\star$ comprises trigger inputs, which are assigned the target-class label t. The poisoning ratio $|\mathcal{D}^\star|/|\mathcal{D}|$ dictates the influence of clean and poisoned data.

Backdoor attacks are of particular interest for CL: as CL-trained models are subsequently used in various downstream

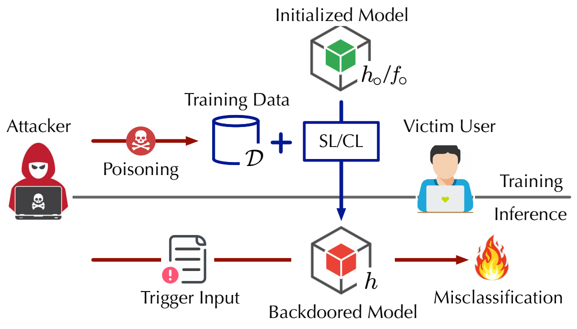

Figure 1: Threat model of TRL.

tasks, backdoor attacks may cause widespread damage [15]. While supervised backdoor attacks are often inapplicable to CL due to their reliance on labels (cf. Eqn (2)), recent studies explore new ways of injecting backdoors into CL-trained models [14, 16, 19, 20].

Specifically, one class of attacks (e.g., SSLBackdoor [14]) hypothesizes that during training, the model learns to recognize a trigger that is only present in the target class and has a rigid, slightly variable shape as a key feature. This enables the model to effectively detect the trigger and accurately predict the target class at inference, even in the absence of other semantic features. The other class of attacks (e.g., CTRL [15]) uses "symmetric alignment": it ensures that the trigger pattern is preserved after augmentations; aligning such augmented makes the trigger pattern appear as semantic features of the target class. Notably, CTRL [15] achieves attack performance comparable to supervised backdoor attacks, suggesting that CL is also highly vulnerable to backdoor attacks.

3 A General Attack Framework

To study supervised and contrastive backdoor attacks side by side, we first define TRL, a general framework that subsumes a number of supervised and contrastive attacks. We begin by introducing its threat model.

3.1 Threat Model

Specifically, we focus on data-level backdoor attacks in the vision domain and assume a threat model similar to prior work [9, 11, 12, 14, 15], as illustrated in Figure 1.

Adversary's Goal: The adversary aims to inject a backdoor into a target model h_θ during training such that at inference time, h_θ classifies trigger inputs to an adversary-defined class t while classifying clean inputs correctly. Formally,

$$\begin{cases} h_\theta(x^\star) = t & x^\star \text{ is triggered} \\ h_\theta(x) = y & x \text{ is clean with ground-truth class } y \end{cases} \tag{3}$$

Adversary's Capability: The adversary corrupts the victim's training data with a small amount of poisoning data $\mathcal{D}^\star$. In the supervised setting, $\mathcal{D}^\star$ comprises input-class pairs $\{(x^\star, t)\}$, in which $x^\star$ is a trigger input; in the contrastive setting, $\mathcal{D}^\star$ only

Algorithm 1: TRL Attack

```
Input: $t$: target class; $k$: number of poisoning inputs; $\tilde{\mathcal{D}}$:
        reference data
Output: $\mathcal{D}^\star$: poisoning data
/* description: select candidate inputs from
   reference data                                   */
/* options: (i) from target class only and (ii)
   across all the classes                           */
1 $\mathcal{D}^\star \leftarrow$ SelectCandidate($\tilde{\mathcal{D}}, t, k$);
2 for $x^\star \in \mathcal{D}^\star$ do
      /* description: apply the trigger to an
         input to generate a trigger input          */
      /* options: (i) universal, (ii) functional,
         and (iii) dynamic trigger                  */
3     update $x^\star \leftarrow$ ApplyTrigger($x^\star$)
4 end
5 return $\mathcal{D}^\star$
```

comprises trigger inputs $\{x^\star\}$ (without labels). The victim's model h_θ is then trained on the union of clean data $\mathcal{D}$ and poisoning data $\mathcal{D}^\star$ (cf. Eqn (2)). Note that such attacks are often practical, even if the victim downloads the training data from credible sources [21].

Adversary's Knowledge: We assume a limited knowledge setting: the adversary has no knowledge about the concrete model h_θ or training strategy (e.g., SimCLR) but has access to a small number of target-class inputs sampled from the same distribution as the training data.

3.2 Overview of TRL

Despite their evident variations, various supervised and contrastive backdoor attacks follow a similar methodology for generating poisoning data, which can be distilled into a high-level framework that we refer to as TRL. As sketched in Algorithm 1, TRL involves two key functions to generate poisoning data: SelectCandidate(), which selects candidate inputs from a reference dataset, and ApplyTrigger(), which applies the trigger to each candidate input to generate the poisoning data.

SelectCandidate() – There are typically two ways of selecting candidate inputs, one across all classes while the other exclusively from the target class.

Among supervised backdoor attacks, dirty-label attacks [9, 11, 12] generate poisoning data along with incorrect labels, while clean-label attacks [22] do not tamper with the labels of poisoning data. Thus, dirty-label attacks select candidates across all classes, while clean-label attacks select candidates from the target class only.

Meanwhile, without data labeling, contrastive backdoor attacks rely on aligning positive pairs to associate the trigger with inputs from the target class (details in § 4). Therefore, contrastive attacks [14, 15] often select candidates from the target class only.

ApplyTrigger() – There are a variety of trigger definitions, including (i) universal triggers (e.g., image patch [9, 12]), (ii) functional triggers (e.g., specific spectral transformations [23]), and (iii) dynamic triggers (e.g., input-aware perturbation generated by a generative model [24]).

3.3 Instantiations

We now discuss the instantiations of TRL in our studies.

To make a fair comparison, in both supervised and contrastive settings, we implement SelectCandidate() as randomly sampling inputs from the target class, corresponding to a clean-label attack.[2]

Further, we instantiate ApplyTrigger() as follows. A universal trigger is defined as a fixed-size (e.g., 5×5) image patch and applied to a pre-defined position (e.g., left lower corner) of the given input. A functional trigger is defined as a specific spectral transformation to the given candidate input (e.g., increasing the magnitude of a particular frequency by a fixed amount). A dynamic trigger is defined as input-specific perturbation generated by a generative model (e.g., GAN). The detailed implementation of these triggers is deferred to § B.

Below we use $\text{TRL}_{\text{SL/CL}}^{\text{uni/fun/dyn}}$ to denote a specific backdoor attack, in which SL/CL indicates supervised/contrastive while uni/fun/dyn indicates the trigger type.

4 Comparison of Supervised and Contrastive Backdoor Attacks

While prior work shows that supervised and contrastive backdoor attacks are comparably effective [15], it remains unclear how they differ fundamentally from a defense perspective. Next, we compare supervised and contrastive attacks in terms of learning dynamics and feature distributions.

4.1 Evaluation Setting

We first describe the setting of our evaluation.

Datasets – We primarily use three benchmark datasets: CIFAR10 [25], which contains 60,000 32×32 images divided into 10 classes; CIFAR100 [25], which includes 100 classes, each containing 600 32×32 images; and ImageNet100 [26], which is a subset of the ImageNet-1K dataset with 100 classes, each with 500 training and 50 testing images.

CL methods – We use three representative CL methods in the vision domain: SimCLR [1], BYOL [2], and MoCo [6].

Attacks – We consider all the instantiations of TRL attacks in § 3.3 including both supervised and contrastive attacks as well as different trigger types (i.e., universal, functional, and dynamic). By default, we allocate 1% of the training data to construct the poisoning dataset $\mathcal{D}^\star$ following Algorithm 1

[2]Note that as the clean-label variant of the dynamic backdoor attack has poor performance, we use its dirty-label variant, which selects inputs across all classes.

Dataset		Attack					
		$\text{TRL}_{\text{SL}}^{\text{fun}}$	$\text{TRL}_{\text{SL}}^{\text{uni}}$	$\text{TRL}_{\text{SL}}^{\text{dyn}}$	$\text{TRL}_{\text{CL}}^{\text{fun}}$ SimCLR	BYOL	MoCo
CIFAR10	ASR (%)	66.7	31.5	99.2	98.3	61.5	89.1
	ACC (%)	82.1	81.4	81.2	82.1	85.6	81.7
CIFAR100	ASR (%)	99.7	83.7	96.4	98.1	82.5	88.8
	ACC (%)	51.4	48.3	50.1	44.2	54.5	46.7
ImageNet100	ASR (%)	98.4	84.2	98.2	42.4	45.1	48.1
	ACC (%)	52.3	52.7	55.3	47.8	49.4	44.3

Table 1. Clean accuracy and attack success rate of different attacks.

while using the remainder as the clean dataset $\mathcal{D}$. We consider class 0 as the target class t. More implementation details about the attacks are deferred to Table 11 in § A.

Models – In both SL and CL, we use ResNet18 [27] as the backbone model. In CL, we add a two-layer MLP projector to map the representations to a 128-dimensional latent space. In § 7, we also discuss the influence of the backbone model.

Metrics – We mainly use two metrics to evaluate the attacks: clean accuracy (ACC) measures the model's accuracy in classifying clean data, and attack success rate (ASR) measures the model's accuracy in classifying poisoning data to the target class.

Table 1 compares the performance of different attacks on benchmark datasets. For contrastive attacks, Table 1 only lists the results of $\text{TRL}_{\text{CL}}^{\text{fun}}$, with other results deferred to § C. Note that for a fair comparison, we set the training epochs to ensure that SL and CL attain similar ACCs. The default parameter setting is listed in § A.

4.2 Learning Dynamics

Given training data as input-class pairs (x,y), SL optimizes the predictive loss (e.g., measured by cross entropy H):

$$\mathcal{L}_{\text{prd}}(\theta) = \mathbb{E}_{(x,y)} H(h_\theta(x), y), \tag{4}$$

where h refers to the end-to-end model $h = g \circ f$. Existing work [28, 29] suggests that in supervised attacks, the backdoor task (i.e., learning the trigger features) is often much "easier" than the benign task (i.e., learning the semantic features). Thus, the model often learns the backdoor task much faster than the benign task, reflected in that the predictive loss of poisoning data drops much faster than that of clean data during training.

Due to the absence of labels, CL optimizes the contrastive loss, which measures the similarity between the variants of the same input under different augmentations (positive pairs) with respect to different inputs (negative pairs). For instance, SimCLR [1] defines the contrastive loss using InfoNCE loss [17] (cf. Eqn (1)), where the numerator term encourages the similarity between positive pair (x,x^+) and the denominator term suppresses the similarity between negative pairs (x,x^-).

To test whether this difference between the learning dynamics of backdoor and benign tasks also holds for contrastive attacks as well, we measure the average contrastive loss of

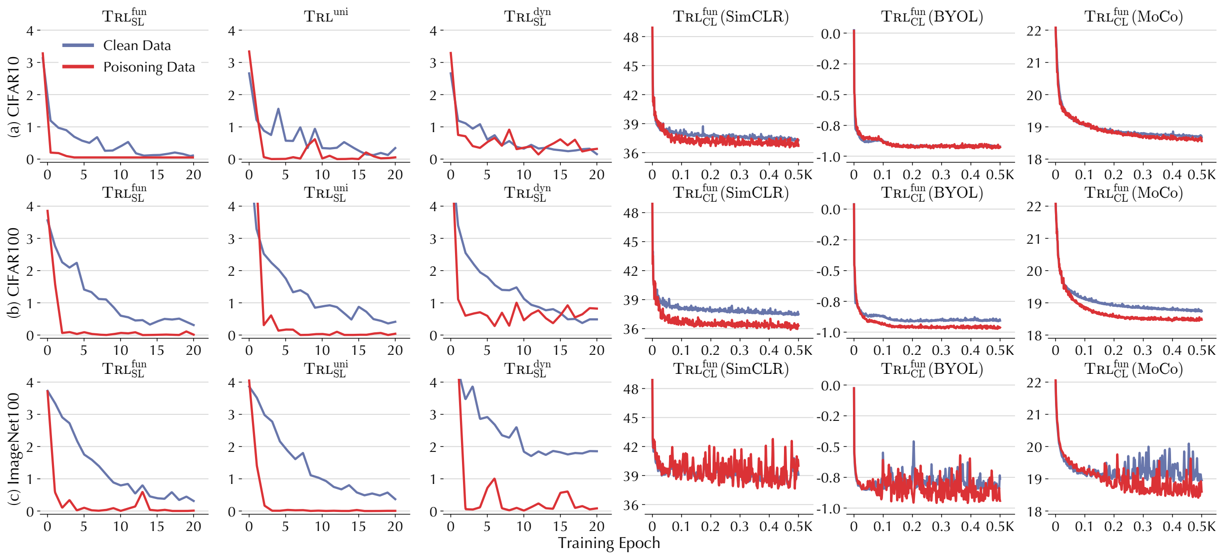
Figure 2: Learning dynamics (measured by the training loss of poisoning and clean data) in supervised and contrastive backdoor attacks on (a) CIFAR10, (b) CIFAR100, and (c) ImageNet100.

poisoning and clean data during training, with results also summarized in Figure 2.

In supervised backdoor attacks, across different trigger settings, the predictive loss of poisoning data decreases faster than that of clean data, which corroborates prior work [28]. For example, the loss of poisoning data in $\text{TRL}_{\text{SL}}^{\text{uni}}$ and $\text{TRL}_{\text{SL}}^{\text{fun}}$ drops sharply to zero in the first few epochs. This difference is especially significant on CIFAR100, which represents a more challenging task in terms of learning semantic features, therefore much more difficult than the backdoor task.

Meanwhile, in contrastive backdoor attacks, across all the CL methods, the training loss of poisoning data decreases gradually at a pace similar to clean data on CIFAR10, CIFAR100, and ImageNet100. While the loss of poisoning data seems slightly lower than clean data on CIFAR100 and ImageNet100 (which may be explained by the more challenging tasks, as indicated by the low clean accuracy in Table 1), their decreasing rates are highly comparable.

Remark 1 – Compared with supervised attacks, in contrastive attacks, the learning dynamics of poisoning and clean data are much less distinguishable.

4.3 Feature Distributions

It is commonly observed in supervised attacks that the poisoning and clean data often form separable clusters in the feature space. This "latent separability" premise is exploited by a number of existing defenses against supervised backdoor attacks [30–34].

To test whether this property also holds for contrastive attacks, we first use t-SNE [35] to visualize the representations of poisoning and clean data in supervised and contrastive attacks, exemplified by $\text{TRL}_{\text{SL}}^{\text{fun}}$ and $\text{TRL}_{\text{CL}}^{\text{fun}}$, with results shown in Figure 3. We have the following observations.

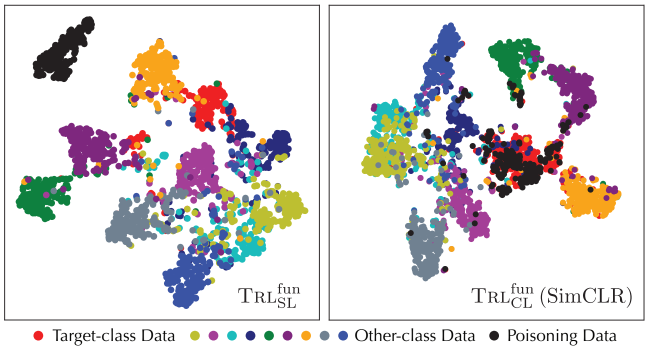

Figure 3: t-SNE visualization of the features of clean and poisoning data in $\text{TRL}_{\text{SL}}^{\text{fun}}$ and $\text{TRL}_{\text{CL}}^{\text{fun}}$ on CIFAR10 (target-class data: red; poisoning data: black).

In supervised attacks, although the clusters of poisoning (in black) and target-class (clean) data (in red) are assigned the same label, they are well separated in the feature space, suggesting that supervised attacks do not necessarily associate the poisoning data with the target-class (clean) data. This finding also corroborates prior work [33]. Meanwhile, in contrastive attacks, the clusters of poisoning and target-class data are highly overlapping in the feature space, suggesting that contrastive attacks may take effect by "entangling" the representations of poisoning and target-class data.

To further validate this observation quantitatively, we define a metric to measure the "entanglement effect" between poisoning and target-class data. Specifically, we define entanglement ratio (ER), which extends the confusion ratio metric [36] to our setting. Specifically, we randomly sample n inputs per class and n poisoning inputs to form the dataset $\mathcal{D}$. In addition, we randomly sample m clean target-class inputs $\mathcal{D}^t$ (disjoint with $\mathcal{D}$) as the testing set. For each $x \in \mathcal{D}^t$, we find its k nearest neighbors $\mathcal{N}_k(f(x))$ among $\mathcal{D}$ in the feature space and then measure the fraction of such neighbors

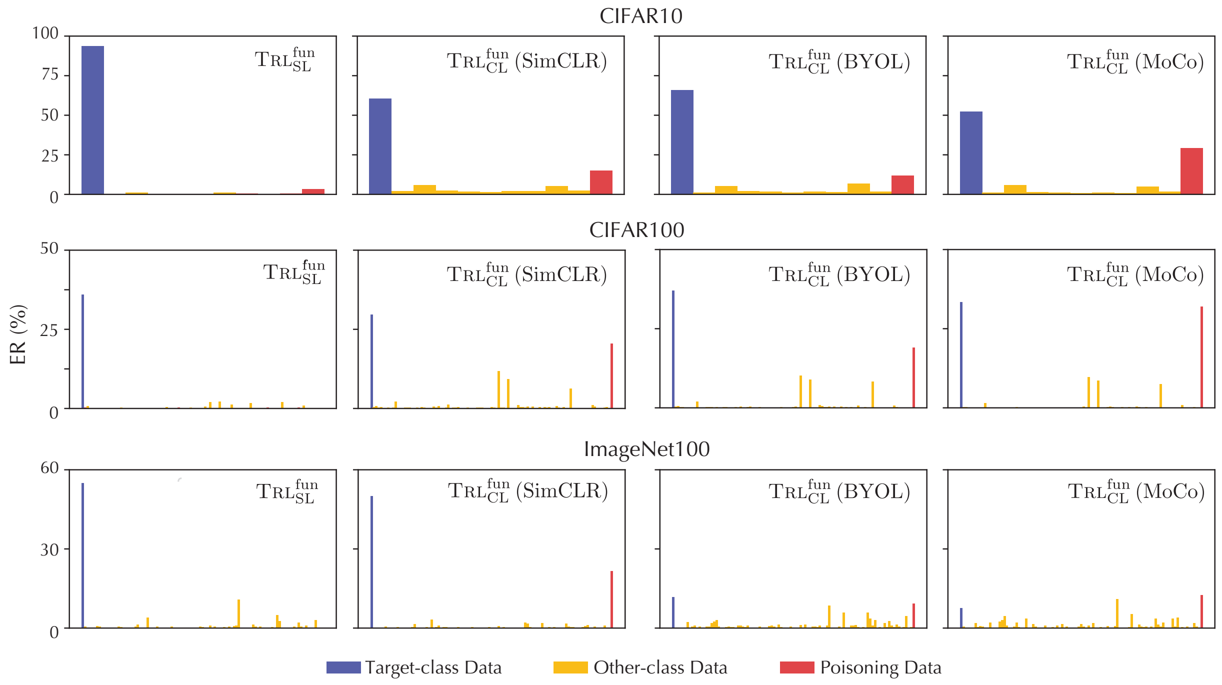

Figure 4: Entanglement ratio of supervised and contrastive backdoor attacks on CIFAR10, CIFAR100 and ImageNet100.

belonging to each class c:

$$\mathrm{ER}(f,c) = \frac{1}{k}\mathbb{E}_{x\in\mathcal{D}^t,x'\in\mathcal{D}}\mathbf{1}_{h(x')=c\wedge f(x')\in\mathcal{N}_k(f(x))} \quad (5)$$

where $\mathbf{1}_p$ is an indicator function that returns 1 if p is true and 0 otherwise and $h(x)$ returns x's class. Consider the poisoning data as a new class $c^\star$. Intuitively, a large $\mathrm{ER}(f,c^\star)$ suggests that the poisoning and clean target-class inputs are tightly entangled in the feature space.

We evaluate the entanglement ratio of different attacks with $n = 800$ and $m = 1,000$. Figure 4 illustrates the degree of entanglement in the feature space between inputs from each class and that from the target class t. Here, each bar represents one distinct class, with the leftmost and rightmost bars corresponding to the target class t and poisoning class $c^\star$, respectively. It is observed that the $\mathrm{ER}(f,c^\star)$ of supervised attacks is not significantly different from any other class $c \neq t$. Meanwhile, in contrastive attacks, $\mathrm{ER}(f,c^\star)$ is much higher than the other classes. For example, for MoCo on CIFAR100, $\mathrm{ER}(f,c^\star)$ is comparable to $\mathrm{ER}(f,t)$ (around 31%), indicating that there is tight entanglement between the poisoning and target-class data.

Remark 2 – Compared with supervised attacks, in contrastive backdoor attacks, the poisoning and clean data are highly entangled in the feature space.

In the evaluation above, we examine the entanglement effect in the feature space (i.e., the output of the last convolutional layer) of the pre-trained encoder. One intriguing question arises: how does this entanglement effect evolve over the model's different layers? To answer this question, we truncate the pre-trained encoder f at a specific ResBlock k at a selected layer l, concatenate its part before (l,k) (denoted by $f_{l,k}$) with a classifier g to form an end-to-end model $h_{l,k}$, and fine-tune g using the downstream dataset.

Truncated Layer l	Truncated Block k	ACC	ASR
3	1	64.5%	15.2%
	2	72.1%	54.2%
4	1	79.3%	82.3%
	2	83.1%	86.4%

Table 2. ACC and ASR of the end-to-end model $h_{l,k}$.

Table 2 summarizes the performance of the end-to-end model $h_{l,k}$, with the encoder trained by TRL$_{SL}^{fun}$ using SimCLR. Observe that $h_{l,k}$'s ACC gradually increases with (l,k). This is expected, as early layers typically extract coarse-grained features, which are then refined throughout subsequent layers. Meanwhile, note that $h_{l,k}$'s ASR also increases with (l,k), suggesting that trigger features and semantic features are not only entangled in the feature space but also intertwined along the feature extraction process.

We further examine the entanglement ratios (cf. Eqn (5)) of the truncated encoder $f_{l,k}$ for different (k,l), with results summarized in Figure 5, following the same setting in Figure 4. Observe that despite the apparent variations of different truncated encoders, the entanglement ratios of different (k,l) show patterns similar to Figure 4, suggesting that the entanglement effect may appear across different layers of the encoder.

Remark 3 – In contrastive backdoor attacks, trigger and semantic features are not only entangled in the feature space but also intertwined throughout the feature extraction process.

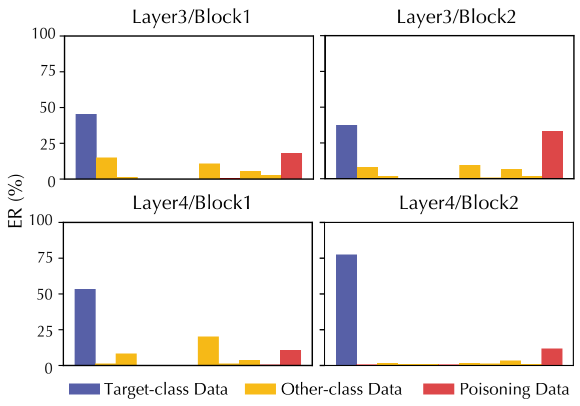

Figure 5: Entanglement ratios of the truncated encoder $f_{l,k}$.

5 Possible Explanations

The empirical study shows that supervised and contrastive backdoor attacks demonstrate distinctive learning dynamics and feature distributions. Below we provide possible explanations for such phenomena, revealing that the two classes of attacks operate through different mechanisms.

5.1 Preliminaries

We begin by introducing two key concepts.

Trigger strength (TS) quantifies the weight of trigger features in poisoning inputs. Specifically, in our setting, we define the trigger strength of $\text{TRL}^{\text{dyn/uni}}_{\text{SL/CL}}$ as the size of the trigger image patch (e.g., 5×5), under a fixed transparency setting. We define the trigger strength of $\text{TRL}^{\text{fun}}_{\text{SL/CL}}$ as the perturbation magnitude in the spectral domain (e.g., increasing the magnitude of a specific frequency band by 50). Intuitively, by adjusting the trigger strength, we balance the importance of trigger features and semantic features in poisoning inputs.

Mutual information (MI) quantifies the amount of information carried by one random variable about another. In our context, we employ MI to estimate the proportion of semantic features retained in the representations of poisoning inputs compared to their corresponding clean inputs. Specifically, consider the representations (i.e., the encoder's outputs) of a clean input and its poisoning counterpart as two random variables X and X', we estimate their MI $I(X;X')$ following [37]. Specifically, let $z_i = (x_i, x'_i)$ be a point sampled from (X, X'). We define the distance between z_i and z_j as

$$\|z_i - z_j\| = \max\{\|x_i - x_j\|_2, \|x'_i - x'_j\|_2\} \tag{6}$$

Let $\varepsilon(i)/2$ be the distance from z_i to its k-th nearest neighbor among the given set of points and $\varepsilon_x(i)/2$ and $\varepsilon_{x'}(i)/2$ be the same distance projected to the X and X' subspaces. Further, let $n_x(i)$ (or $n_{x'}(i)$ be the number of points with distance to x_i (or x'_i) less than $\varepsilon_x(i)/2$ (or $\varepsilon_{x'}(i)/2$). Then, the MI of X and X' is quantified by:

$$I(X;X') = \psi(k) - \frac{1}{k} - \langle \psi(n_x) + \psi(n_{x'}) \rangle + \psi(N) \tag{7}$$

where ψ is the digamma function, k is a hyper-parameter, and N is the number of points in X (or X'). We use Eqn (7) to measure the proportion of semantic features retained in the representations of poisoning inputs compared to their corresponding clean inputs. In the implementation, we set $N = 2,000$ and $k = 5$.

5.2 Supervised Backdoor Attacks

Under the supervised setting, the backdoor attack is implemented as optimizing the following objective function:

$$\min_{\theta} \mathcal{L}_{\text{prd}}(\theta) + \lambda \mathcal{L}^{\star}_{\text{prd}}(\theta) \tag{8}$$

where $\mathcal{L}$ and $\mathcal{L}^{\star}$ denote the loss function defined in Eqn (4), measured on clean data $\mathcal{D}$ and poisoning data $\mathcal{D}^{\star}$ respectively, and the hyper-parameter λ balances the influence of $\mathcal{D}$ and $\mathcal{D}^{\star}$ (e.g., via controlling $|\mathcal{D}^{\star}|/|\mathcal{D}|$). Intuitively, the first and second terms represent the benign task (i.e., learning semantic features) and backdoor task (i.e., learning trigger features), respectively. We hypothesize that in supervised attacks, if the trigger feature is sufficiently evident, the learning of benign and backdoor tasks tends to occur independently at the data level; that is, it focuses on learning the backdoor task from the poisoning data and focuses on learning the benign task from the clean data.

To validate this hypothesis, we assess the effect of trigger strength on the ASR, ACC, and MI of supervised backdoor attacks, with results shown in Figure 6. Observe that as the trigger strength varies from 0 to 1K, the ASR of $\text{TRL}^{\text{fun}}_{\text{SL}}$ increases rapidly and remains around 100%, while its MI quickly drops and stays at the minimum level. This observation suggests that in successful attacks, the model focuses on learning trigger features (backdoor task) only from the poisoning data. Meanwhile, the trigger strength has little impact on the ACC, indicating that the model effectively learns the benign task from the clean data. Similar trends are also observed on other supervised attacks in § 7.1.

Remark 4 – In supervised backdoor attacks, the learning of benign and backdoor tasks occurs independently at the data level, with a focus on learning trigger features from poisoning data and semantic features from clean data.

The independence of benign and backdoor tasks in supervised attacks explains the empirical observations in § 4. In terms of learning dynamics, as the trigger feature (e.g., a specific image patch) is often simpler than the varying semantic features, the model learns the backdoor task much faster than the benign task. In terms of feature distributions, as Eqn (8) does not specify any constraints on the latent representations of poisoning and clean data, although associated with the same class, the representations of poisoning and target-class data are not necessarily proximate in the feature space.

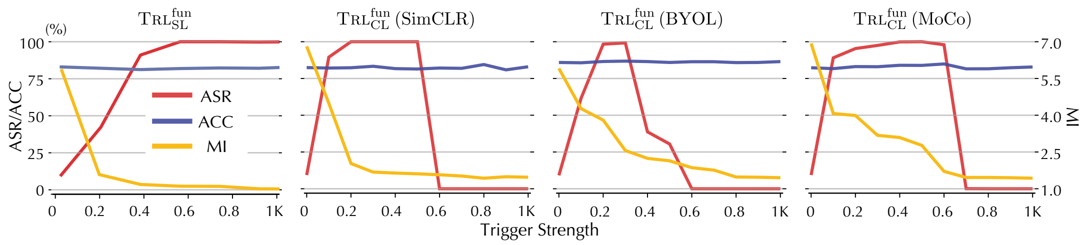

Figure 6: ASR, ACC, and MI of different attacks with respect to the trigger strength (measured by the magnitude of spectral perturbation).

5.3 Contrastive Backdoor Attacks

Meanwhile, under the contrastive setting, the backdoor attack is implemented as the following objective function:

$$\min_{\theta} \mathcal{L}_{\text{ctr}}(\theta) + \lambda \mathcal{L}_{\text{ctr}}^{\star}(\theta) \tag{9}$$

where $\mathcal{L}_{\text{ctr}}$ and $\mathcal{L}_{\text{ctr}}^{\star}$ are the contrastive loss defined in Eqn (1), measured on $\mathcal{D}$ and $\mathcal{D}_{\star}$ respectively. Because there are no labels, contrastive attacks manipulate the distributions of poisoning and target-class data in the feature space. To do so, the model must learn both trigger and semantic features from the poisoning data, so that the semantic features intertwine poisoning data with target-class data, while the trigger features connect all poisoning data.

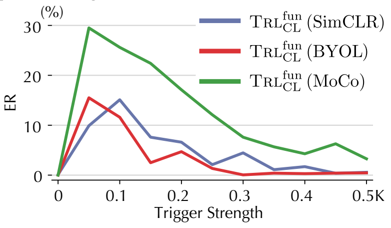

Figure 7: Entanglement effect with respect to trigger strength.

To confirm our analysis, we assess the impact of trigger strength on the ASR, ACC, and MI of contrastive backdoor attacks, with results shown in Figure 6. Across all the CL methods, as the trigger strength varies from 0 to 1K, the ASR of $\text{TRL}_{\text{CL}}^{\text{fun}}$ first increases to 100% and then drops quickly, while its MI gradually decreases during the process. The finding suggests that the optimal attack is attained only if the semantic features are partially retained in the representations of poisoning data. This can be explained by the fact that if the trigger features are insignificant, the semantic features dominate the poisoning data, resulting in weak connections among poisoning data; meanwhile, if the trigger features dominate the poisoning data, the weak semantic features negatively impact the entanglement between poisoning and target-class data. To validate this explanation, we further measure the entanglement between poisoning and target-class data under varying trigger strength. As shown in Figure 7, the entanglement ratio is not a monotonic function of the trigger strength but follows a similar trend as the ASRs in Figure 6. We can thus conclude:

Remark 5 – In contrastive backdoor attacks, the learning of benign and backdoor tasks is intertwined in the poisoning data.

This interdependence between benign and backdoor tasks in contrastive backdoor attacks easily explains the empirical observations in § 4. In terms of learning dynamics, because the model learns both benign and backdoor tasks (the latter is simpler than the former) from poisoning data, the contrastive loss of poisoning data tends to decrease at a rate similar to clean data. In terms of feature distributions, as the model extracts both trigger and semantic features from poisoning data, it naturally intertwines poisoning and target-class data in the feature space.

6 Defense Implications

We now explore the implications of the unique characteristics of contrastive attacks from a defense perspective.

6.1 Learning Dynamics-Based Defenses

We first examine defenses premised on the difference between the learning dynamics of benign and backdoor tasks.

Anti-Backdoor Learning (ABL) [28] is a filtering defense that segregates poisoning data and unlearns backdoored models. Intuitively, as the backdoor task is simpler than the benign task, assuming the loss of poisoning data drops quickly in early training epochs while the loss of clean data decreases at a steady pace, ABL detects poisoning data and unlearns the backdoored model by applying gradient descent on the clean data and gradient ascent on the detected poisoning data.

Despite its effectiveness against supervised attacks, it is challenging to extend ABL to CL because the learning dynamics of poisoning data in contrastive attacks are much less distinctive (cf. § 4.2). We empirically validate this by applying ABL in detecting the poisoning data of $\text{TRL}_{\text{SL}}^{\text{fun}}$ and $\text{TRL}_{\text{CL}}^{\text{fun}}$ under the same setting as in § 4.1. We follow the hyper-parameter setting in [28]. Specifically, if the loss of any training input goes below a threshold $\gamma = 0.5$, we activate gradient ascent to boost its loss to γ. In the CL setting, we set γ according to a fixed ratio compared to the largest loss drop: $(\gamma - \ell_{\text{final}})/(\ell_{\text{init}} - \ell_{\text{final}})$, where ℓ_{init} and ℓ_{final} respectively denote the loss of the training input before and after training. Table 3 reports ABL's performance under the poisoning rate fixed as 1%. The false positive rate (FPR) measures the fraction of clean inputs falsely detected as poisoning, the true positive rate (TPR) measures the fraction of poisoning inputs correctly identified, and the isolation rate denotes the percentage of samples isolated by ABL. A lower FPR or a higher TPR indicates more accurate detection.

Isolation Rate	$\textsc{Trl}^{\text{fun}}_{\text{CL}}$		$\textsc{Trl}^{\text{fun}}_{\text{SL}}$	
	TPR	FPR	TPR	FPR
1%	2.4%	0.99%	92.8%	0.07%
5%	39.6%	4.8%	99.4%	4.1%
10%	56.0%	9.5%	99.8%	9.1%
20%	63.6%	19.6%	99.8%	19.2%

Table 3. ABL against $\textsc{Trl}^{\text{fun}}_{\text{SL}}$ and $\textsc{Trl}^{\text{fun}}_{\text{CL}}$ on CIFAR10.

Observe that ABL effectively detects the poisoning data of $\textsc{Trl}^{\text{fun}}_{\text{SL}}$ but is much less effective against $\textsc{Trl}^{\text{fun}}_{\text{CL}}$. For example, with 1% isolation rate, ABL detects 92.8% of the poisoning data of $\textsc{Trl}^{\text{fun}}_{\text{SL}}$, which drops to 2.4% against $\textsc{Trl}^{\text{fun}}_{\text{CL}}$. The increase of TPR against $\textsc{Trl}^{\text{fun}}_{\text{CL}}$ with the isolation rate is mainly attributed to the low poisoning rate (1%). We thus conclude that ABL is ineffective against contrastive attacks, which confirms the observations in § 4.2. This is due to the intertwined learning dynamics of both benign and backdoor tasks. We defer the evaluation of ABL against $\textsc{Trl}^{\text{uni}}_{\text{SL/CL}}$ to § C.2.

6.2 Feature Distribution-Based Defenses

The entanglement between the representations of poisoning and clean data also causes challenges for defenses that rely on the separability of poisoning data in the feature space.

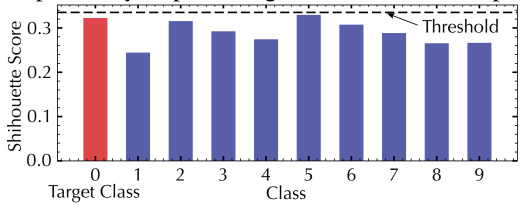

Figure 8: AC against $\textsc{Trl}^{\text{fun}}_{\text{CL}}$ (SimCLR) on CIFAR10.

Activation clustering (AC) [31] is a data inspection method. It trains the model using the potentially poisoning data and collects the penultimate-layer activation of each input. AC assumes the poisoning data in the target class forms a separate cluster that is either small or far from the class center. It identifies the target class by calculating the silhouette score of each class, with a higher score indicating a stronger fit to two clusters. Additionally, since the attacker is assumed to be unable to poison more than half of the data, the smaller cluster is considered to be the poisoning data. Assuming the data labels are available, we evaluate the effectiveness of AC against both supervised and contrastive attacks. We set the hyper-parameters according to the original paper.

Figure 8 shows that AC fails to identify the target class (class 0), which has a lower score compared to other classes (e.g., class 5), not to mention detecting the poisoning data. This may be attributed to the tight entanglement between the representations of poisoning and clean data.

Statistical contamination analyzer (SCAn) [33] detects poisoning data based on the statistical properties of its representations. Specifically, it applies an EM algorithm to decompose an input into two terms: identity and variance; it then analyzes the representations in each class and identifies the target class most likely to be characterized by a mixture model. To evaluate SCAn against contrastive attacks, following [33], we use 1,000 inputs randomly sampled from the testing set to build the decomposition model, which we apply to analyze 5,000 poisoning and 5,000 clean inputs. For comparison, we also evaluate SCAn against the supervised attack $\textsc{Trl}^{\text{fun/uni}}_{\text{SL}}$. Similar to STRIP [38], we vary the FPR under three different FPR settings: 0.5%, 1.0%, and 2.0%, and evaluate the TPR.

FPR	TPR			
	$\textsc{Trl}^{\text{fun}}_{\text{CL}}$	$\textsc{Trl}^{\text{fun}}_{\text{SL}}$	$\textsc{Trl}^{\text{uni}}_{\text{CL}}$	$\textsc{Trl}^{\text{uni}}_{\text{SL}}$
0.5%	28.0%	63.0%	28.4%	51.5%
1.0%	28.0%	66.5%	63.3%	71.8%
2.0%	28.0%	68.0%	76.7%	85.5%

Table 4. SCAn against $\textsc{Trl}^{\text{fun}}_{\text{CL}}$ (SimCLR) and $\textsc{Trl}^{\text{fun}}_{\text{SL}}$on CIFAR10.

As shown in Table 4, compared with the supervised attack, SCAn is much less effective, especially for $\textsc{Trl}^{\text{fun}}_{\text{CL}}$. For instance, SCAn detects over 63.0% of the poisoning data of $\textsc{Trl}^{\text{fun}}_{\text{SL}}$, which drops to 28.0% against $\textsc{Trl}^{\text{fun}}_{\text{CL}}$. This can be attributed to the entanglement between the representations of poisoning and clean data, while SCAn relies on the separability of poisoning data in the feature space.

Fine-pruning (FP) [39] assumes poisoning and clean data activate different neurons and sanitizes backdoored models by pruning neurons that are dormant with respect to clean data. Specifically, it measures the average activation of each neuron with respect to clean data from a validation set and prunes the set of least active neurons. Following [39], we apply FP on the penultimate layer of the backdoored model. In [39], it keeps pruning the model until the tolerance of accuracy (5%) reduction is reached. For a more detailed analysis, we show the ASR and ACC with respect to the variation of the pruned channels from 0 to 500.

Figure 9 shows the ASR and ACC of each attack as a function of the pruning rate of FP. Observe that FP effectively defends against supervised attacks. By pruning about 240 neurons, the ASR of $\textsc{Trl}^{\text{fun}}_{\text{SL}}$ is reduced to zero while incurring an acceptable ACC drop. Meanwhile, the ASR and ACC of contrastive attacks decrease at a similar pace. To attain effective defenses, it is necessary to prune almost all the neurons. This further suggests the tight entanglement between poisoning and clean data in the feature space.

Remark 6 – The unique characteristics of contrastive backdoor attacks render defenses based on learning dynamics and feature distributions ineffective.

6.3 Downstream Defenses

Thus far, we have demonstrated the challenges of defending against contrastive backdoor attacks based on either learning dynamics or feature distributions. Next, we consider a more diverse set of defenses applied to the end-to-end model $h =$

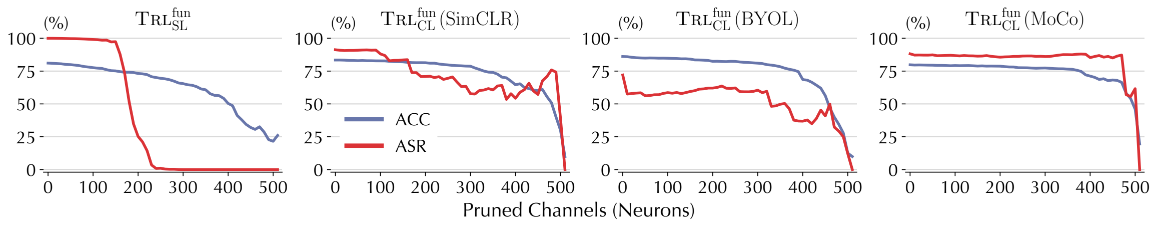

Figure 9: FP against $\text{TRL}_{\text{CL}}^{\text{fun}}$ and $\text{TRL}_{\text{SL}}^{\text{fun}}$ on CIFAR10.

	$\text{TRL}_{\text{CL}}^{\text{fun}}$	$\text{TRL}_{\text{SL}}^{\text{fun}}$	$\text{TRL}_{\text{CL}}^{\text{uni}}$	$\text{TRL}_{\text{SL}}^{\text{uni}}$
Anomaly Index	0.72 ± 0.38	1.47 ± 0.54	1.09 ± 0.49	4.57 ± 1.28

Table 5. Effectiveness of NC on CIFAR10.

Detection Threshold	FPR	TPR
0.56	0.5%	0.5%
0.65	1.0%	1.1%
0.75	2.0%	2.0%

Table 6. STRIP against $\text{TRL}_{\text{CL}}^{\text{fun}}$ (SimCLR) on CIFAR10.

$g \circ f$ that comprises the backdoored encoder f and a classifier g in the downstream task.

NeuralCleanse (NC) [40] is a model inspection method to detect backdoors. Specifically, for each class, NC reverse-engineers the "minimal" trigger (measured by the L_1-norm of its binary mask) required to misclassify clean inputs from all other classes into this class. It then runs outlier detection and marks any class with a trigger significantly smaller than the other classes as an outlier and infected. We evaluate NC's efficacy in both supervised and contrastive learning. In the contrastive setting, an end-to-end model, in which the encoder trained by $\text{TRL}_{\text{CL}}^{\text{fun}}$ or $\text{TRL}_{\text{CL}}^{\text{uni}}$ using SimCLR, is fine-tuned using clean data. In the supervised setting, the model is directly trained by $\text{TRL}_{\text{SL}}^{\text{fun}}$ or $\text{TRL}_{\text{SL}}^{\text{uni}}$.

As shown in Table 5, NC failed to detect any backdoors in the contrastive setting. For example, for $\text{TRL}_{\text{CL}}^{\text{fun}}$, the anomaly index of the target class varies around 0.72, much lower than the detection threshold of 2. This is attributed to the presence of functional triggers and aggressive data augmentation methods, which make it challenging for NC to reverse-engineer a stable trigger. In contrast, NC successfully detects backdoors against supervised attacks, especially $\text{TRL}_{\text{SL}}^{\text{uni}}$.

STRIP [38] is an inference-time data inspection method. Specifically, it assumes that the prediction of poisoning data is insensitive to perturbation due to the dominance of trigger features, whereas the prediction of clean data may vary greatly with perturbation. For an incoming input, STRIP intentionally perturbs it (e.g., superimposing various image patterns) and utilizes the randomness of its prediction to determine whether it is triggered. As it is a run-time detection method, we use an end-to-end model (consisting of the backdoored encoder and downstream classifier) to evaluate its effectiveness. Following [38], we first estimate the entropy distribution of clean inputs on a validation set, select an entropy detection threshold with a given FPR (0.5%, 1.0%, and 2.0%), and remove all training inputs with entropy below this threshold.

As shown in Table 6, STRIP is ineffective in detecting the

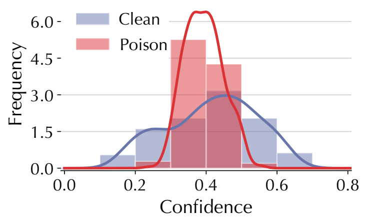

Figure 10: Distributions of the prediction confidence of poisoning and clean data.

poisoning data of $\text{TRL}_{\text{CL}}^{\text{fun}}$. For instance, with FRR fixed as 1.0%, the TPR of STRIP is as low as 1.1%, suggesting that STRIP fails to distinguish poisoning and clean data. This may also be explained by the entanglement effect: as the poisoning and clean data share similar representations, the perturbation to them leads to similar measures. The results on $\text{TRL}_{\text{CL/SL}}^{\text{uni}}$ are deferred to § C.3.

Backdoor defense via decoupling (BDD) [41] is designed to mitigate the impact of poisoning inputs that tend to cluster together in the feature space of compromised DNNs. This approach consists of three stages: (1) the backbone is trained in a self-supervised manner, which aligns inputs with the same ground-truth label in the feature space; (2) the remaining classifier is trained in a standard manner, using all labeled training inputs; and (3) low-confidence inputs are removed based on the end-to-end model.

Clearly, (1) and (2) share the same settings as the threat model of TRL. Thus, we focus on (3) to evaluate the effectiveness of BDD. The distributions of the classification confidence of poisoning and clean (from the target class) inputs are shown in Figure 10. Note that poisoning inputs not only have similar classification confidence as clean inputs but also exhibit lower variance. This makes it more challenging to establish a confidence threshold to distinguish between them. Consequently, using BDD mechanisms to defend against TRL tends to be ineffective.

Fine-tuning [42] is a method to adapt a pre-trained encoder to a specific downstream task by fine-tuning a classifier. Prior work [42] shows that fine-tuning may mitigate backdoor attacks to a certain extent. In the previous evaluation, following [14], we focus on scenarios where the downstream and pre-training datasets are identical. Here, we explore cases in which these datasets differ, such as pre-training the encoder on CIFAR10 and fine-tuning the classifier on CIFAR100. Since the datasets may have varied class structures, we focus on untargeted attacks and measure the attack effectiveness by the model's accuracy drop in classifying trigger inputs compared to clean inputs.

Table 7 summarizes the results. Observe that the contrastive backdoor attack substantially affects the model's performance, regardless of the setting of pre-training and downstream datasets. For instance, when the encoder is pre-trained on CIFAR100 using SimCLR and then adapted to CIFAR10, the model achieves 53.3% accuracy and 18.1% accuracy (i.e., 35.2% accuracy drop) on clean and trigger inputs, respectively. This observation indicates that fine-tuning using downstream datasets may not mitigate contrastive backdoor attacks. Additionally, we observe that most trigger inputs are misclassified into a single incorrect class. We speculate that these trigger inputs are clustered together and located close to a specific class in the feature space, which is also close to the target class in the pre-training dataset.

Pre-training / Downstream Dataset	Method	ACC (Clean)	ACC Drop (Poisoning)
CIFAR10 / CIFAR100	Supervised	15.1%	7.9%
	SimCLR	30.1%	26.2%
	BYOL	32.1%	23.4%
	MoCo	28.3%	23.6%
CIFAR100 / CIFAR10	Supervised	46.7%	32.4%
	SimCLR	53.3%	35.2%
	BYOL	60.7%	44.3%
	MoCo	55.7%	41.4%

Table 7. Performance of fine-tuning on $\text{TRL}_{\text{CL}}^{\text{fun}}$.

Remark 7 – The existing defenses are not easily retrofitted to defend against contrastive backdoor attacks in the setting of downstream tasks.

7 Discussion

Thus far, we uncover the fundamental differences between supervised and contrastive backdoor attacks, propose possible explanations, and reveal the important implications entailed by such differences from a defense perspective. Next, we further explore a few key questions: (i) are our findings impacted by other factors (e.g., model architectures and trigger definitions)? (ii) given the inadequacy of existing defenses, are there any promising alternatives? (iii) what are the limitations of this work?

7.1 Other Factors

We explore whether our findings are influenced by other factors, including model architectures and trigger definitions.

Model Architectures – To evaluate the influence of backbone model architectures, besides ResNet18, we also consider three other popular DNN architectures (inlcuding ResNet50 [27], ShuffleNet [43], and MobileNet [44]) and train a SimCLR encoder on CIFAR10 as outlined in § 4.

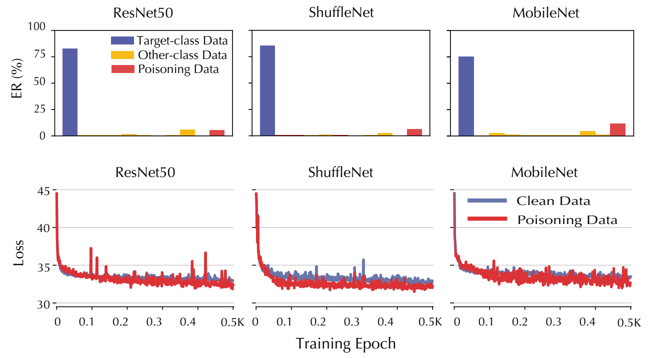

Figure 11: Learning dynamics and entanglement ratio under different backbone architectures.

Figure 11 shows the learning dynamics and entanglement ratio under varying backbone architectures. Our results indicate that: (i) the loss-decreasing pace is comparable for both poisoning and clean inputs across different architectures, and (ii) the poisoning inputs and the clean target-class inputs are entangled in the feature space under all the settings. The findings suggest that the backbone architectures have a limited impact on contrastive backdoor attacks.

Alternative triggers – In the previous evaluation, for contrastive attacks, we primarily focus on the functional trigger $\text{TRL}_{\text{CL}}^{\text{fun}}$. Here, we investigate whether the trigger definition impacts our findings. Specifically, we consider the universal trigger $\text{TRL}_{\text{CL}}^{\text{uni}}$ as an alternative trigger definition. Following [14], we specify a 5×5 image patch as the trigger pattern, which is randomly placed at a random location of a given input.

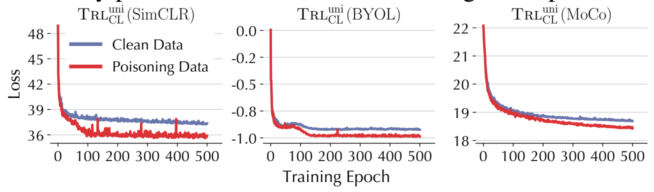

Figure 12: Learning dynamics of $\text{TRL}_{\text{CL}}^{\text{uni}}$.

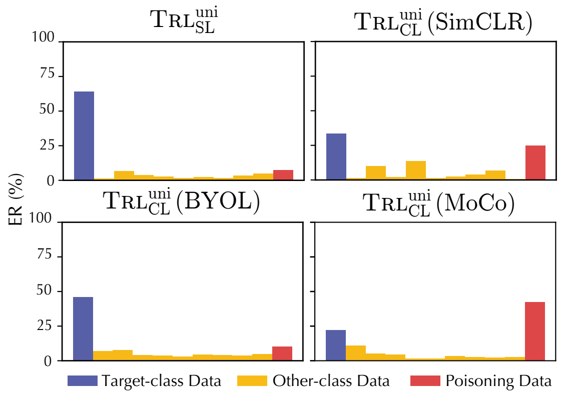

Figure 13: Entanglement ratios of $\text{TRL}_{\text{SL/CL}}^{\text{uni}}$ on CIFAR10.

Figure 12 shows the learning dyanmics of $\text{TRL}_{\text{CL}}^{\text{uni}}$. Similar to $\text{TRL}_{\text{CL}}^{\text{fun}}$ (cf. Figure 2), $\text{TRL}_{\text{CL}}^{\text{uni}}$ also exhibits distinct learning dynamics compared with supervised attacks. Specifically, the contrastive loss of both poisoning and clean data decreases

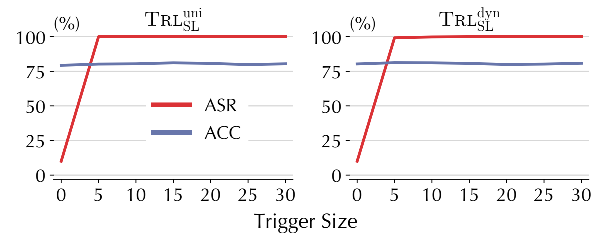

Figure 14: Attack performance of $\text{TRL}_{\text{SL}}^{\text{uni}}$ and $\text{TRL}_{\text{SL}}^{\text{dyn}}$ with respect to trigger strength (image patch size).

CL Method	Trigger Size				
	1	5	10	15	20
SimCLR	9.8%	57.0%	0.1%	0.0	0.0
BYOL	10.7%	34.6%	65.6%	2.01%	0.3%
MoCo	9.7%	45.6%	64.3%	18.1%	0.8%

Table 8. Attack effectiveness of $\text{TRL}_{\text{CL}}^{\text{uni}}$ with respect to trigger size.

gradually at fairly similar paces. Further, Figure 13 evaluates the entanglement ratios of $\text{TRL}_{\text{SL/CL}}^{\text{uni}}$. Observe that similar to Figure 4, the entanglement effect is also more evident in the contrastive setting. Thus, the difference in entanglement effect is rather trigger-agnostic but stems from the underlying learning paradigms.

Further, we conduct experiments to examine the influence of trigger strength (measured by the image patch size) on the effectiveness of supervised and contrastive attacks. Figure 14 shows the ASR and ACC of $\text{TRL}_{\text{SL}}^{\text{uni}}$ and $\text{TRL}_{\text{SL}}^{\text{dyn}}$ as a function of trigger strength, which complements the results of Figure 6. Observe that in both cases of universal and dynamic triggers, as the trigger strength increases, the ASR of supervised attacks quickly peaks and remains around 100%; meanwhile, the trigger strength has little impact on the ACC, indicating the independence of backdoor and benign tasks at the data level for supervised attacks.

In contrast, as shown in Table 8, across different CL methods, as the trigger size increases from 1 to 20, the effectiveness of $\text{TRL}_{\text{CL}}^{\text{uni}}$ initially increases and then decreases abruptly, similar to the trend observed about $\text{TRL}_{\text{CL}}^{\text{fun}}$ in Figure 6.

Remark 8 – The unique characteristics of contrastive backdoor attacks are agnostic to the concrete trigger definitions and backbone architectures.

7.2 Potential Defenses

We demonstrate that due to the specificities of contrastive backdoor attacks, defenses that attempt to segregate poisoning data based on learning dynamics and feature separability tend to fail. Given such limitations, we explore alternative defense strategies and discuss their potential challenges to defend against contrastive attacks.

Density-based Filtering – We have an interesting observation in our evaluation: the trigger inputs tend to form a cluster in the feature space, which is often much denser than the clusters formed by clean inputs. For example, we use a SimCLR model trained on CIFAR10 and measure the average pairwise

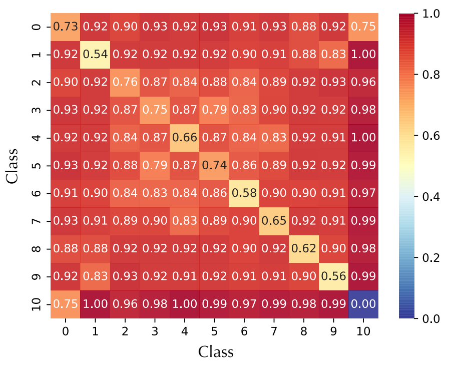

Figure 15: Average normalized pair-wise L_2 distance on CIFAR10 (trigger inputs: class 10; clean inputs: class 0 - 9). Diagonal cells represent intra-class distance while off-diagonal cells represent inter-class distance.

L_2 distance between inputs in the feature space. Figure 15 shows the normalized intra-class and inter-class distance. Observe that trigger inputs (class 10, with normalized distance 0.0) tend to cluster much more tightly compared with clean inputs (class 0 - 9, with normalized distance ≥ 0.56).

Motivated by this observation, we propose a density-based filtering defense. After training the encoder on potentially poisoning data, we use it to generate features for all inputs. Next, we apply a density-based clustering method (e.g., OPTICS [45] or DBSCAN [46]) to cluster the training inputs based on their features. The densest clusters are identified as potentially poisoning data and subsequently removed from the training data.

Below we evaluate this approach in defending against $\text{TRL}_{\text{CL}}^{\text{fun}}$ (SimCLR) on CIFAR10 under two distinct poisoning rates: 1% and 5%. Specifically, we employ DBSCAN as the clustering algorithm to sift out poisoning inputs. DBSCAN hinges on two key parameters: (i) Minimum samples $N_{\min}$, which is the minimum number of inputs required to form a cluster. Intuitively, a larger $N_{\min}$ results in denser clusters. Given that the number of poisoning inputs is typically small, we set $N_{\min}$ to either 30 or 100 in the evaluation. (ii) Maximum distance ε, which thresholds the distance between two inputs to determine whether they belong to the same cluster. To determine ε, we calculate the average distance between each input and its $K = N_{\min}$ nearest neighbors, plot the average K-distances in ascending order on a K-distance graph (as shown in Figure 16), and set ε as the point of maximum curvature, where the graph has the largest decreasing rate. We set $\varepsilon = 0.3$ in the evaluation.

Table 9 summarizes the effectiveness of this density-based filtering measured by true positive rate (TPR) and false positive rate (FPR), as well as the attack success rate (ASR) and clean accuracy (ACC) of the model trained on the post-filtering training data. Notably, density-based filtering effectively sifts out a majority of poisoning inputs across various settings. For instance, under the setting of 5% poisoning rate, $\varepsilon = 0.3$, and $N_{\min} = 30$, it successfully filters 98.8% of the poisoning inputs. Further, observe that after filtering potentially

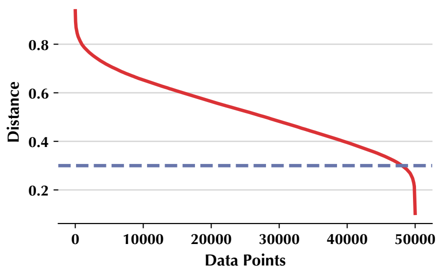

Figure 16: K-distance graph

Parameter		Poisoning Rate	Filtering		Re-training	
ε	$N_{\min}$		TPR	FPR	ASR	ACC
0.3	30	1%	96.6%	10.2%	13.3%	80.1%
	100		80.1%	0.4%	43.8%	81.8%
	30	5%	98.8%	9.8%	19.9%	77.8%
	100		97.8%	0.4%	59.8%	78.1%

Table 9. Effectiveness of density-based filtering defense.

poisoning data and retraining the model, we may not only effectively mitigate the attack but also retain the model's utility. For instance, it achieves 13.3% ASR and 80.1% ACC under 1% poisoning rate and $N_{\min} = 30$. However, the effectiveness of re-training seems sensitive to the parameter setting. For instance, under 1% poisoning rate and $N_{\min} = 100$, as it fails to filter out approximately 100 poisoning inputs, the attack is not mitigated effectively.

We further evaluate the effectiveness of density-based filtering with OPTICS as the underlying clustering algorithm, which is less sensitive to the parameter setting. It achieves an impressive 99.6% TPR and 0% FPR under 1% poisoning rate. However, it has a much larger execution overhead. For example, on our platform, OPTICS requires 1.5 hours to complete the filtering, whereas DBSCAN finishes in approximately 10 seconds. Thus, we consider enhancing both the effectiveness and efficiency of density-based filtering defense as our ongoing work.

Data-free pruning – In addition to filtering potentially poisoning data, we also explore in-depth defenses for scenarios where filtering proves ineffective. We have shown that data-dependent pruning [39] is insufficient to mitigate contrastive backdoor attacks, due to the indistinguishable feature patterns of clean and poisoning data (§ 6.2). Surprisingly, data-free pruning, which is believed to be less effective than data-dependent pruning, shows promising performance against contrastive backdoor attacks.

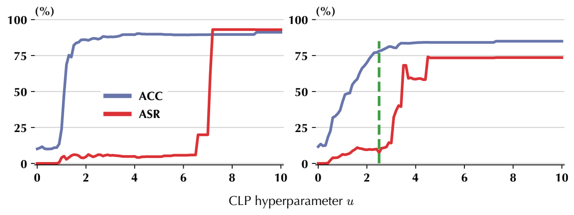

Figure 17: CLP against $\text{TRL}_{\text{SL}}^{\text{fun}}$ and $\text{TRL}_{\text{CL}}^{\text{fun}}$ on CIFAR10.

Specifically, we consider channel Lipschitzness-based pruning (CLP) [47], a data-free defense to remove backdoors. It exploits the observation that a subset of channels is more sensitive to trigger features compared to other channels, while the sensitivity of each channel can be estimated by the upper bound of its channel Lipschitz constant (UCLC), which can be computed in a data-free manner. Thus, given the mean (μ_k) and variance (σ_k) of the UCLCs in the k-th convolutional layer, CLP prunes channels with UCLCs larger than $\mu_k + u\sigma_k$, where u is a hyper-parameter.

We apply CLP on models backdoored by $\text{TRL}_{\text{SL}}^{\text{fun}}$ and $\text{TRL}_{\text{CL}}^{\text{fun}}$ with varying u. As shown in Figure 17, by properly setting u, it is possible to effectively reduce the ASR of $\text{TRL}_{\text{CL}}^{\text{fun}}$ with limited ACC drop (about 5%). However, compared with $\text{TRL}_{\text{SL}}^{\text{fun}}$, the proper range of u for $\text{TRL}_{\text{CL}}^{\text{fun}}$ is much narrower ([2.2, 3.0] versus [1.8, 6.5]). To address this challenge, we propose to empirically set u as the knee point of the ACC curve. Specifically, we measure the ACC under varying u, apply the Savitzky–Golay filter to smooth the curve [48], and identify the knee point of the smoothed curve as the optimal u during pruning. By applying this approach, we identify the optimal u as 2.4 in the contrastive setting, as indicated by the green dashed line in Figure 17, which well balances utility and defensive efficacy (80.4% ACC and 14.3% ASR).

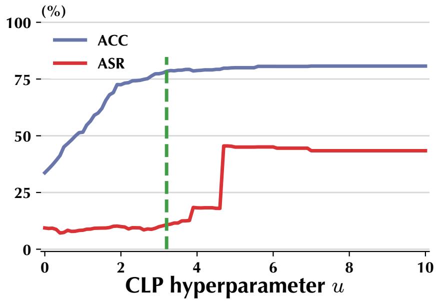

Figure 18: CLP against the retrained model

Ensemble defenses – We further combine the previous two defenses to form a powerful ensemble defense against contrastive backdoor attacks. Specifically, we first apply density-based filtering to filter potentially poisoning data, retrain the model on the post-filtering training data, and then apply data-free pruning on the retrained model. We simulate the setting that the filtering hyperparameter is not properly set (e.g., $N_{\min} = 100$ under 1% poisoning rate), which results in a retrained model with 43.8% ASR and 81.8% ACC. By applying data-free pruning on this retrained model, as illustrated in Figure 18, we further reduce ASR to 12.6% while retaining ACC around 79.6%. We believe this ensemble defense represents a promising direction worthy of further research.

7.3 Limitations and Future Work

We further discuss the limitations of this work and point to several future directions. (i) **Key properties:** our study primarily focuses on the differences between supervised and contrastive backdoor attacks, as reflected in learning dynamics and feature distributions. Other properties (e.g., neuron activation patterns) might also vary across these attack types. Future research could explore these underexplored properties to develop more effective defenses. (ii) **Alternative defenses:** our study highlights the defense implications of key differences between supervised and contrastive backdoor attacks. We illustrate the challenges faced by several representative defenses against contrastive backdoor attacks. Comprehensively investigating and benchmarking all the SOTA defenses (e.g., [49, 50]) against contrastive backdoor attacks is a valuable avenue for future research. (iii) **Data modality:** while our study mainly focuses on the vision domain, contrastive learning has been extended to other domains such as natural language processing and multimodal learning [51, 52]. It is worth investigating how domain constraints (e.g., discrete perturbation, semantic preservation, and interference between different modalities) affect contrastive backdoor attacks and their potential implications. (iv) **Generalization to self-supervised learning (SSL):** while focusing on contrastive learning methods, our findings might extend to the wider SSL paradigm (e.g., masked autoencoder [53]). It is essential, however, to validate these implications across broader contexts. Future work should explore the applicability and adaptability of our insights to alternative SSL frameworks and examine the implications for defenses in these new settings.

In summary, while limited in several aspects, this work represents a solid step towards understanding and defending against the emerging threat of backdoor attacks in contrastive learning. Our findings highlight the need for defenses tailored to the specificities of contrastive backdoor attacks.

8 Additional Related work

In addition to the aforementioned related work, we further survey the literature most relevant to this work.

Contrastive learning – Recent years have witnessed the striding advances of contrastive learning (CL) [1, 2, 4]. Compared with supervised learning (SL), CL obviates the reliance on data labeling and still learns high-quality representations from complex data, thereby facilitating various downstream tasks. Meanware, the popularity of CL also spurs intensive research on its security properties. Existing work explores the adversarial robustness of CL [54, 55]. It is shown that, as a nice side effect, obviating the reliance on data labeling may benefit the model's robustness to adversarial examples, label corruption, and common data corruptions [56]. However, whether this robustness benefit also generalizes to other types of attacks remains an open question.

Supervised Contrastive learning – Supervised contrastive learning (Sup-CL) [57] introduces label information in the contrastive learning framework. By aligning latent representations with task-specific class semantics, Sup-CL enhances the quality of learned representations. Recent work [58, 59] applies Sup-CL in malware analysis and attains notable performance gains. Compared with Sup-CL, CL operates without explicit labels, relying on data augmentations to form and discern positive and negative pairs in an unsupervised manner. Despite the differences, the attacks in this work can be easily adapted to binary classification (e.g., malware analysis) under both CL and Sup-CL settings. Specifically, by injecting the trigger pattern into a small number of goodware samples during training, both self-supervised and full-supervised variants of CL may inadvertently associate this pattern with goodware, as they seek to minimize the distance between similar entities in the latent space. At inference time, any malware with this pattern may be misclassified as goodware. We consider comparing the attacks under CL and Sup-CL settings in practical settings (e.g., malware analysis) as our ongoing work.

Backdoor attacks – As a major threat to machine learning security, backdoor attacks inject malicious backdoors into the victim's model during training and activate such backdoors at inference. Many backdoor attacks have been proposed for SL, which can be categorized along (i) attack targets – input-specific [60], class-specific [33] or any-input [9], (ii) attack vectors – polluting training data [12] or releasing backdoored models [61], and (iii) optimization metrics – attack effectiveness [11], transferability [13], or attack evasiveness [60, 62–64].

Backdoor attacks are of particular interest for CL, potentially causing widespread damage due to their extensive application in downstream tasks. Supervised backdoor attacks, dependent on data labeling, are often inapplicable to CL, prompting innovative new approaches. For instance, BadEncoder [19] injects backdoors into pre-trained encoders and releases backdoored models to victims, while SSLBackdoor [14] uses image patch triggers to poison data; PoisonedEncoder [16] poisons the training data by randomly combining target inputs with reference inputs. Recently, CTRL [15] utilizes spectral triggers, achieving attack performance comparable to supervised attacks.

Yet, despite numerous studies on supervised and contrastive backdoor attacks, their fundamental differences remain unexplored. This work bridges this gap by revealing the distinctive mechanisms behind these attacks.

Backdoor defenses – To mitigate the threats of backdoor attacks, many defenses have been proposed, which can be categorized according to their strategies [65]: (i) input filtering, which purges poisoning examples from training data [31, 66]; (ii) model inspection, which determines whether a given model is backdoored and, if so, recovers the target class and the potential trigger [40, 67–69]; and (iii) input inspection, which detects trigger inputs at inference time [33, 38, 70, 71].

However, mainly designed for supervised backdoor attacks, the effectiveness of these defenses in the CL setting remains unclear, raising several critical questions: can the existing defenses be retrofitted to CL attacks? If not, what new challenges do these attacks entail? How can we address such challenges? This work systematically explores these key questions.

9 Conclusion

In this study, we examine the fundamental distinctions between supervised and contrastive backdoor attacks. Using a unified attack framework, we uncover that these attacks operate through different mechanisms, resulting in distinct learning dynamics and feature distributions. More importantly, we show that the unique characteristics of contrastive backdoor attacks entail important implications, requiring new and tailored defenses. Our findings shed new light on developing more robust contrastive learning techniques and point to several promising directions for further research.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 1951729, 1953813, 2119331, and 2212323. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. This work was partly supported by the National Key Research and Development Program of China under No. 2022YFB3102100.

References

[1] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” in *Proceedings of IEEE Conference on Machine Learning (ICML)*, 2020.

[2] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar *et al.*, “Bootstrap your own latent - a new approach to self-supervised learning,” in *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[3] A. Newell and J. Deng, “How useful is self-supervised pretraining for visual tasks?” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[4] X. Chen and K. He, “Exploring simple siamese representation learning,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[5] X. Chen, H. Fan, R. Girshick, and K. He, “Improved baselines with momentum contrastive learning,” *ArXiv e-prints*, 2020.

[6] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, “Momentum contrast for unsupervised visual representation learning,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[7] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, “Hierarchical text-conditional image generation with clip latents,” *ArXiv e-prints*, 2022.

[8] OpenAI, “Dalle2,” https://openai.com/dall-e-2/.

[9] T. Gu, B. Dolan-Gavitt, and S. Garg, “Badnets: Identifying vulnerabilities in the machine learning model supply chain,” *ArXiv e-prints*, 2017.

[10] W. Brendel, J. Rauber, and M. Bethge, “Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models,” *ArXiv e-prints*, 2017.

[11] R. Pang, H. Shen, X. Zhang, S. Ji, Y. Vorobeychik, X. Luo, A. Liu, and T. Wang, “A tale of evil twins: Adversarial inputs versus poisoned models,” in *Proceedings of ACM Conference on Computer and Communications (CCS)*, 2020.

[12] Y. Liu, S. Ma, Y. Aafer, W.-C. Lee, J. Zhai, W. Wang, and X. Zhang, “Trojaning attack on neural networks,” in *Proceedings of Network and Distributed System Security Symposium (NDSS)*, 2018.

[13] Y. Yao, H. Li, H. Zheng, and B. Y. Zhao, “Latent backdoor attacks on deep neural networks,” in *Proceedings of ACM Conference on Computer and Communications (CCS)*, 2019.

[14] A. Saha, A. Tejankar, S. A. Koohpayegani, and H. Pirsiavash, “Backdoor attacks on self-supervised learning,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[15] C. Li, R. Pang, Z. Xi, T. Du, S. Ji, Y. Yao, and T. Wang, “An embarrassingly simple backdoor attack on self-supervised learning,” in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2023.

[16] H. Liu, J. Jia, and N. Z. Gong, “Poisonedencoder: Poisoning the unlabeled pre-training data in contrastive learning,” in *Proceedings of USENIX Security Symposium (SEC)*, 2022.

[17] A. van den Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” *ArXiv e-prints*, 2018.

[18] S. Li, B. Z. Hao Zhao, J. Yu, M. Xue, D. Kaafar, and H. Zhu, “Invisible backdoor attacks against deep neural networks,” *ArXiv e-prints*, 2019.

[19] J. Jia, Y. Liu, and N. Z. Gong, “Badencoder: Backdoor attacks to pre-trained encoders in self-supervised learning,” in *Proceedings of IEEE Symposium on Security and Privacy (S&P)*, 2021.

[20] N. Carlini and A. Terzis, “Poisoning and backdooring contrastive learning,” in *Proceedings of International Conference on Learning Representations (ICLR)*, 2022.

[21] N. Carlini, M. Jagielski, C. A. Choquette-Choo, D. Paleka, W. Pearce, H. Anderson, A. Terzis, K. Thomas, and F. Tramèr, “Poisoning web-scale training datasets is practical,” *ArXiv e-prints*, 2023.

[22] A. Shafahi, W. R. Huang, M. Najibi, O. Suciu, C. Studer, T. Dumitras, and T. Goldstein, “Poison frogs! targeted clean-label poisoning attacks on neural networks,” in *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[23] Y. Li, Y. Jiang, Z. Li, and S.-T. Xia, “Backdoor learning: A survey,” *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[24] T. A. Nguyen and A. Tran, “Input-aware dynamic backdoor attack,” in *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[25] A. Krizhevsky and G. Hinton, “Learning Multiple Layers of Features from Tiny Images,” *Technical report, University of Toronto*, 2009.

[26] Y. Le and X. Yang, “Tiny imagenet visual recognition challenge,” *CS 231N*, vol. 7, no. 7, p. 3, 2015.

[27] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[28] Y. Li, X. Lyu, N. Koren, L. Lyu, B. Li, and X. Ma, “Anti-backdoor learning: Training clean models on poisoned data,” in *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[29] H. Wang, J. Hong, A. Zhang, J. Zhou, and Z. Wang, “Trap and Replace: Defending Backdoor Attacks by Trapping Them into an Easy-to-Replace Subnetwork,” in *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[30] T. Wang, Y. Yao, F. Xu, S. An, H. Tong, and T. Wang, “An invisible black-box backdoor attack through frequency domain,” in *Proceedings of European Conference on Computer Vision (ECCV)*, 2022.

[31] B. Chen, W. Carvalho, N. Baracaldo, H. Ludwig, B. Edwards, T. Lee, I. Molloy, and B. Srivastava, “Detecting backdoor attacks on deep neural networks by activation clustering,” *ArXiv e-prints*, 2018.

[32] K. Huang, Y. Li, B. Wu, Z. Qin, and K. Ren, “Backdoor Defense via Decoupling the Training Process,” in *Proceedings of International Conference on Learning Representations (ICLR)*, 2022.

[33] D. Tang, X. Wang, H. Tang, and K. Zhang, “Demon in the variant: Statistical analysis of dnns for robust backdoor contamination detection,” in *Proceedings of USENIX Security Symposium (SEC)*, 2020.

[34] J. Hayase, W. Kong, R. Somani, and S. Oh, “Spectre: defending against backdoor attacks using robust statistics,” in *Proceedings of IEEE Conference on Machine Learning (ICML)*, 2021.

[35] L. Van der Maaten and G. Hinton, “Visualizing data using t-sne,” *Journal of Machine Learning Research*, vol. 9, no. 11, 2008.

[36] Y. Wang, Q. Zhang, Y. Wang, J. Yang, and Z. Lin, “Chaos is a ladder: A new theoretical understanding of contrastive learning via augmentation overlap,” *ArXiv e-prints*, 2022.

[37] A. Kraskov, H. Stögbauer, and P. Grassberger, “Estimating mutual information,” *Physical review E*, vol. 69, no. 6, p. 066138, 2004.

[38] Y. Gao, C. Xu, D. Wang, S. Chen, D. C. Ranasinghe, and S. Nepal, “Strip: A defence against trojan attacks on deep neural networks,” in *Proceedings of Annual Computer Security Applications Conference (ACSAC)*, 2019.

[39] K. Liu, B. Dolan-Gavitt, and S. Garg, “Fine-pruning: Defending against backdooring attacks on deep neural networks,” in *Proceedings of International Symposium on Research in Attacks, Intrusions, and Defenses (RAID)*, 2018.

[40] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, and B. Y. Zhao, “Neural cleanse: Identifying and mitigating backdoor attacks in neural networks,” in *Proceedings of IEEE Symposium on Security and Privacy (S&P)*, 2019.

[41] K. Huang, Y. Li, B. Wu, Z. Qin, and K. Ren, “Backdoor defense via decoupling the training process,” in *Proceedings of International Conference on Learning Representations (ICLR)*, 2021.

[42] Y. Liu, Y. Xie, and A. Srivastava, “Neural trojans,” in *Proceedings of IEEE International Conference on Computer Design (ICCD)*, 2017.

[43] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, “Shufflenet v2: Practical guidelines for efficient cnn architecture design,” in *Proceedings of European Conference on Computer Vision (ECCV)*, 2018.

[44] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[45] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, “Optics: Ordering points to identify the clustering structure,” in *Proceedings of ACM International Conference on Management of Data (SIGMOD)*, 1999.

[46] A. Ram, S. Jalal, A. S. Jalal, and M. Kumar, “A density based algorithm for discovering density varied clusters in large spatial databases,” *International Journal of Computer Applications*, vol. 3, no. 6, pp. 1–4, 2010.

[47] R. Zheng, R. Tang, J. Li, and L. Liu, “Data-free Backdoor Removal based on Channel Lipschitzness,” in *Proceedings of European Conference on Computer Vision (ECCV)*, 2022.

[48] W. H. Press and S. A. Teukolsky, “Savitzky-golay smoothing filters,” *Computers in Physics*, vol. 4, no. 6, pp. 669–672, 1990.

[49] X. Xu, Q. Wang, H. Li, N. Borisov, C. A. Gunter, and B. Li, “Detecting ai trojans using meta neural analysis,” in *Proceedings of IEEE Symposium on Security and Privacy (S&P)*, 2021.

[50] M. Weber, X. Xu, B. Karlaš, C. Zhang, and B. Li, “Rab: Provable robustness against backdoor attacks,” in *Proceedings of IEEE Symposium on Security and Privacy (S&P)*, 2023.

[51] S. Lee, D. Bok Lee, and S. J. Hwang, “Contrastive learning with adversarial perturbations for conditional text generation,” in *Proceedings of International Conference on Learning Representations (ICLR)*, 2021.

[52] J. Giorgi, O. Nitski, B. Wang, and G. Bader, “Declutr: Deep contrastive learning for unsupervised textual representations,” in *Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL)*, 2021.

[53] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, “Masked autoencoders are scalable vision learners,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[54] Z. Jiang, T. Chen, T. Chen, and Z. Wang, “Robust pre-training by adversarial contrastive learning,” in *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[55] L. Fan, S. Liu, P.-Y. Chen, G. Zhang, and C. Gan, “When does contrastive learning preserve adversarial robustness from pre-training to finetuning?” in *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[56] D. Hendrycks, M. Mazeika, S. Kadavath, and D. Song, “Using self-supervised learning can improve model robustness and uncertainty,” in *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[57] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, “Supervised contrastive learning,” in *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[58] L. Yang, W. Guo, Q. Hao, A. Ciptadi, A. Ahmadzadeh, X. Xing, and G. Wang, “{CADE}: Detecting and explaining concept drift samples for security applications,” in *Proceedings of USENIX Security Symposium (SEC)*, 2021.

[59] Y. Chen, Z. Ding, and D. Wagner, “Continuous learning for android malware detection,” *ArXiv e-prints*, 2023.

[60] A. Shafahi, W. Ronny Huang, M. Najibi, O. Suciu, C. Studer, T. Dumitras, and T. Goldstein, “Poison frogs! targeted clean-label poisoning attacks on neural networks,” in *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[61] Y. Ji, X. Zhang, S. Ji, X. Luo, and T. Wang, “Model-Reuse Attacks on Deep Learning Systems,” in *Proceedings of ACM SAC Conference on Computer and Communications (CCS)*, 2018.

[62] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, “Targeted backdoor attacks on deep learning systems using data poisoning,” *ArXiv e-prints*, 2017.

[63] A. Turner, D. Tsipras, and A. Madry, “Label-consistent backdoor attacks,” *ArXiv e-prints*, 2019.

[64] S. Zhao, X. Ma, X. Zheng, J. Bailey, J. Chen, and Y.-G. Jiang, “Clean-label backdoor attacks on video recognition models,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[65] R. Pang, Z. Zhang, X. Gao, Z. Xi, S. Ji, P. Cheng, and T. Wang, “Trojanzoo: Towards unified, holistic, and practical evaluation of neural backdoors,” in *Proceedings of IEEE European Symposium on Security and Privacy (Euro S&P)*, 2020.

[66] B. Tran, J. Li, and A. Madry, “Spectral signatures in backdoor attacks,” in *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[67] S. Kolouri, A. Saha, H. Pirsiavash, and H. Hoffmann, “Universal litmus patterns: Revealing backdoor attacks in cnns,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[68] X. Huang, M. Alzantot, and M. Srivastava, “Neuroninspect: Detecting backdoors in neural networks via output explanations,” *ArXiv e-prints*, 2019.

[69] Y. Liu, W.-C. Lee, G. Tao, S. Ma, Y. Aafer, and X. Zhang, “Abs: Scanning neural networks for back-doors by artificial brain stimulation,” in *Proceedings of ACM Conference on Computer and Communications (CCS)*, 2019.

[70] M. Subedar, N. Ahuja, R. Krishnan, I. J. Ndiour, and O. Tickoo, “Deep probabilistic models to detect data poisoning attacks,” *ArXiv e-prints*, 2019.

[71] Z. Xi, T. Du, C. Li, R. Pang, S. Ji, J. Chen, F. Ma, and T. Wang, “Defending pre-trained language models as few-shot learners against backdoor attacks,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.

A Parameter Setting

Table 10 lists the default parameter setting for training the encoder using different CL methods and SL models. Table 11 lists the default parameter setting of different attacks in our evaluation.

Parameter	CL Method			SL
	SimCLR	BYOL	MoCo	
Optimizer	SGD	SGD	SGD	SGD
Learning rate	0.06	0.06	0.06	0.1
Optimizer momentum	0.9	0.9	0.9	0.9
Momentum	-	0.996	0.999	-
Weight decay	1e-4	1e-4	1e-4	1e-4
Epochs	500	500	500	20
Batch size	512	512	512	512
Temperature	0.5	-	0.5	-
Moving average	-	0.996	-	-
Memory size	-	-	65536	-

Table 10. Parameter setting of encoder training.

Parameter	Attack Type		
	$\text{TRL}_{\text{CL}}^{\text{fun}}$ / $\text{TRL}_{\text{SL}}^{\text{fun}}$	$\text{TRL}_{\text{CL}}^{\text{uni}}$ / $\text{TRL}_{\text{SL}}^{\text{uni}}$	$\text{TRL}_{\text{SL}}^{\text{dyn}}$
Poison ratio	1%	1%	1%
Target class	0	0	0
Trigger size	-	5×5	32×32
Magnitude	100.0	-	-
Block size	32	-	-
Frequency bands	15, 31	-	-
Trigger position	-	random	0.1
Generator training epochs	-	-	10
Backdoor prob. ρ_a	-	-	0.1
Cross-trigger prob. ρ_b	-	-	0.1

Table 11. Default parameter setting of attacks.

B Implementation Details

B.1 Different Variants of TRL

We elaborate on the implementation of different variants of TRL.

$\text{TRL}_{\text{CL}}^{\text{fun}}$ [15] – The SelectCandidate function samples candidates from a specified class. The ApplyTrigger function modifies the input image by converting it from RGB to YCbCr format, dividing it into blocks (e.g., 32×32), and applying the discrete cosine transform (DCT) to shift from the spatial to the frequency domain. It then increases the magnitude of selected high-frequency bands by 50, reverses the DCT to return to the spatial domain, and finally converts the image back from YCbCr to RGB.

$\text{TRL}_{\text{CL}}^{\text{uni}}$ [14] – SelectCandidate samples candidates from the target class. ApplyTrigger follows [14]: it first generates a random 5×5 image patch as the trigger and then applies it to a randomly selected position of the given candidate input.

$\text{TRL}_{\text{SL}}^{\text{fun}}$ [15] – SelectCandidate samples candidates from the target class. ApplyTrigger is the same as $\text{TRL}_{\text{CL}}^{\text{fun}}$. In addition, the labels of all the candidate inputs are set as the target class.

$\text{TRL}_{\text{SL}}^{\text{uni}}$ [9, 12] – SelectCandidate samples candidates from the target class. ApplyTrigger generates a 5×5 image patch as the trigger and applies it to a random position of the given candidate. In addition, the labels of all the candidates are set as the target class. For a fair comparison, we let $\text{TRL}_{\text{SL}}^{\text{uni}}$ and $\text{TRL}_{\text{CL}}^{\text{uni}}$ share the same poisoning data.

$\text{TRL}_{\text{SL}}^{\text{dyn}}$ [24] – SelectCandidate samples candidates across all the classes. ApplyTrigger generates input-specific triggers using a generative model G. Specifically, the training process runs in three modes: i) with probability $\rho_a \in (0,1)$, it runs in the "backdoor" mode in which G produces a trigger $G(x)$ and applied it to given input x, while its label is perturbed to the target class; ii) with probability $\rho_b \in (0,1)$, it runs in the "cross-trigger" mode, in which G produces the trigger $G(x)$ and applies it to a different input x', while its original label is preserved; iii) with probability $1-\rho_a-\rho_b$, it runs in the "clean" mode, in which x is used as a clean input.

B.2 Defense Details

We detail the implementation of various defenses in our evaluation.

ABL [28] – We follow the settings in their original paper. Specifically, we set the loss threshold $\gamma = 0.5$. If the loss of a training example goes below γ, gradient ascent will be activated to boost its loss to γ; otherwise, the loss stays the same. In the contrastive learning setting, we set the threshold γ according to the same ratio compared to the largest loss.

AC [31] – We set the hyper-parameters as the original paper.

STRIP [38] – In the original paper, it estimates the entropy distribution of clean samples on a validation set, selects an entropy threshold with a given FPR, and eventually removes all training samples with entropy below this threshold. In our work, we measure its effectiveness under three FPR settings: 0.5%, 1.0%, and 2.0%.

SCAn [33] – Similar to STRIP [38], we vary the FPR under three settings: 0.5%, 1.0%, and 2.0%, and evaluate the TPR.

FP [39] – In the original paper, it keeps pruning the model until the tolerance of accuracy (5%) reduction is reached. In our work, for a more detailed analysis, we show the ASR and ACC with respect to the variation of the pruned channels from 0 to 500.

C Additional Experiments

C.1 Attack effectiveness of $\text{TRL}_{\text{CL}}^{\text{fun}}$

In § 7.1, we evaluate the universal trigger $\text{TRL}_{\text{CL}}^{\text{uni}}$ as an alternative trigger definition. Following [14], we specify a 5×5 image patch as the trigger pattern, which is randomly placed at a random location of a given sample. Table 12 shows the clean accuracy and the corresponding attack success rate.

Dataset		Attack		
		$\text{TRL}_{\text{CL}}^{\text{fun}}$		
		SimCLR	BYOL	MoCo
CIFAR10	ASR (%)	33.2	49.2	53.1
	ACC (%)	79.4	84.3	80.6

Table 12. Clean accuracy and attack success rate of $\text{TRL}_{\text{CL}}^{\text{fun}}$.

C.2 ABL against $\text{TRL}_{\text{SL}}^{\text{uni}}$ and $\text{TRL}_{\text{CL}}^{\text{uni}}$

Table 13 shows the effectiveness of ABL against $\text{TRL}_{\text{SL}}^{\text{uni}}$ and $\text{TRL}_{\text{CL}}^{\text{uni}}$. Observe that ABL is much less effective in the contrastive setting compared to the supervised setting.

Isolation Rate	$\text{TRL}_{\text{CL}}^{\text{uni}}$		$\text{TRL}_{\text{SL}}^{\text{uni}}$	
	TPR	FPR	TPR	FPR
1%	0.8%	1.0%	93.6%	0.06%
5%	24.3%	4.8%	99.6%	4.0%
10%	32.2%	9.8%	99.8%	9.1%
20%	43.6%	19.7%	99.8%	19.2%

Table 13. ABL against $\text{TRL}_{\text{SL}}^{\text{uni}}$ and $\text{TRL}_{\text{CL}}^{\text{uni}}$ on CIFAR10.

C.3 STRIP against $\text{TRL}_{\text{SL}}^{\text{uni}}$ and $\text{TRL}_{\text{CL}}^{\text{uni}}$

Table 14 reveals a divergent defensive capacity of STRIP against distinct attacks: it encounters discernible difficulties when countering the contrastive backdoor attack, $\text{TRL}_{\text{CL}}^{\text{uni}}$, as evidenced by a consistently low TPR at a designated FPR. In contrast, STRIP demonstrates a better defense against the supervised attack, $\text{TRL}_{\text{SL}}^{\text{uni}}$, especially its dirty label variant, exhibiting a pronounced increase in TPR at a specified FPR.

Attack	Decision Threshold	FPR	TPR
$\text{TRL}_{\text{CL}}^{\text{uni}}$	5.11	0.5%	0.5%
	5.4	1.0%	0.8%
	5.5	2.0%	1.5%
$\text{TRL}_{\text{SL}}^{\text{uni}}$	4.17	0.5%	0.6%
	5.3	1.0%	2.4%
	5.4	2.0%	5.9%
$\text{TRL}_{\text{SL}}^{\text{uni}}$ (dirty)	5.11	0.5%	0.6%
	5.21	1.0%	30.6%
	5.37	2.0%	59.8%

Table 14. STRIP against $\text{TRL}_{\text{SL}}^{\text{uni}}$ and $\text{TRL}_{\text{CL}}^{\text{uni}}$ on CIFAR10.

C.4 Poisoning in Fine-tuning

Figure 19 illustrates the learning dynamics of additional poisoning in the fine-tuning stage on CIFAR10. First, a backdoored encoder is obtained using $\text{TRL}_{\text{SCL}}^{\text{fun}}$ (SimCLR). Subsequently, the encoder and the classifier head are fine-tuned on a small dataset, which includes 50 clean examples from each class, along with an additional 50 poisoning examples specifically from the target class. Throughout the fine-tuning stage, we continuously monitor the loss of both clean and poisoning data. In Figure 19, we observe that the loss of poisoning data decreases much more rapidly than that of clean data, which is consistent with the supervised learning setting in § 4.2. This is explained by that fine-tuning is inherently a form of supervised learning.

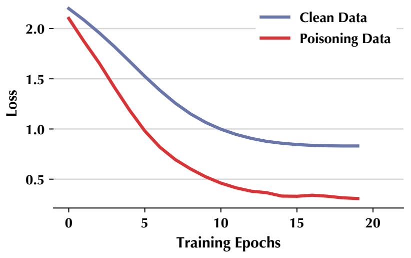

Figure 19: Learning dynamics of additional poisoning in fine-tuning.