DIFT: Protecting Contrastive Learning Against Data Poisoning Backdoor Attacks

Jiang Zhu[1], Yulin Jin[1], Qingqing Ye[1], Zhibiao Guo[1], Kun Fang[1], Ruochen Du[2], Yingnan Zhao[2,3], Haibo Hu[1,4*]

[1]Department of Electrical and Electronic Engineering, The Hong Kong Polytechnic University
[2]School of Computer Science and Technology, Harbin Engineering University
[3]National Engineering Laboratory for Modeling and Emulation in E-Government, Harbin Engineering University
[4]Research Centre for Privacy and Security Technologies in Future Smart Systems, The Hong Kong Polytechnic University
polyu-jiang.zhu@connect.polyu.hk, jyl990903@163.com, qqing.ye@polyu.edu.hk, zhibiao.guo@connect.polyu.hk, kun.fang@polyu.edu.hk, {drc96, zhaoyingnan}@hrbeu.edu.cn, haibo.hu@polyu.edu.hk

Abstract

Contrastive learning (CL) is a popular learning paradigm that excels in extracting meaningful representations from unlabeled data. Recent studies have shown that CL is highly vulnerable to backdoor attacks. Current defenses against backdoor attacks in CL are primarily reactive and post-training. That is, the detection and elimination of backdoors are executed in the deployment phase of a given well-trained model. However, these post-training defenses are usually prone to degrading model utility and resource-intensive, causing that the backdoor detection and elimination from a fully-trained model is quite challenging. To address this issue, we argue for a fundamental perspective, i.e., integrating the defense into the model's training phase, and propose a novel framework to mitigate the backdoor in CL, namely Density-Based Identification and Fine-Tuning (DIFT). Specifically, DIFT identifies potential poisoned samples during the early training phase via detecting embeddings with abnormal poisoning characteristic in the feature space. Then, to remove backdoors and preserve model utility, the detected poisoned samples are leveraged to fine-tune the model, and the remaining clean samples are further involved into training the model after the fine-tuning. DIFT, as a proactive training-time defense, avoids the problematic backdoor removal and the high computational cost associated with those reactive post-training methods. We empirically evaluate DIFT on various CL algorithms against backdoor attack. Experimental results demonstrate that our method exhibits promising defense effectiveness while maintaining model's clean data accuracy.

Introduction

Extracting meaningful representations from data is a key task in training deep neural networks (DNNs) (Goodfellow, Bengio, and Courville 2016; Bengio, Courville, and Vincent 2013; Zhang et al. 2025; Bai et al. 2025; Xiao et al. 2025; Li et al. 2025; Tang et al. 2024). Contrastive learning (CL) (Chen and He 2021; He et al. 2020; Chen et al. 2020; Grill et al. 2020) is a self-supervised learning paradigm that trains on large, unlabeled datasets. It learns compact representations by encoding and comparing data samples. This involves pushing of positive samples, which are typically derived from the same data with different data augmentation, closer together in the feature space, and negative samples, which are different data points from the training dataset, further apart. Meanwhile, the increasing popularity of CL raises significant security concerns (Wang, Zhu, and Gao 2024; Li et al. 2024; Liu, Jia, and Gong 2022; Zheng et al. 2024; Feng et al. 2023; Zhang et al. 2024b; Liang et al. 2024). A representative example is that the training of the encoder faces the threat from backdoor attacks (Adi et al. 2018; Bagdasaryan and Shmatikov 2021; Gao et al. 2019; Jia, Liu, and Gong 2022; Li et al. 2021a; Wang et al. 2025, 2024). In such attacks, the training data of the encoder is poisoned with a specific, attacker-defined trigger, causing the trained encoder to behave normally on clean data but to exhibit malicious behavior when presented with the trigger. The backdoored encoder would further pose potential security risks to the downstream applications. Therefore, exploring the robustness of CL against backdoor attacks is of great significance (Li et al. 2024).

In response to this threat, a large portion of research has focused on post-training defense mechanisms. (Li et al. 2024) highlights significant challenges on CL backdoor defense. It explores the feasibility of many existing backdoor defense schemes, among them Lipschitzness-based pruning (Zheng et al. 2022) is a promising solution for backdoor removal. It also proposes leveraging density-based clustering algorithms, such as DBSCAN (Ester et al. 1996) and OPTICS (Ankerst et al. 1999), to identify poisoned training data. After detecting the poisoned data, it retrains the model to remove backdoor influence. To further enhance defense, the authors propose an ensemble approach that combines pruning with filtering-retraining to achieve more effective results. However, the post-training filtering methods are often rudimentary, relying exclusively on unsupervised clustering algorithms and lacking a calibrated design. This inherent limitation results in either a suboptimal true positive rate or a significant increase in computational time. As a result, it is challenging to balance practicality with high detection effectiveness.DeDe (Hou, Li, and Yao 2024) addresses the challenge of detecting stealthy backdoor by reconstructing embeddings. This method exploits the characteristic that data containing trigger encodes different semantic information to clean data. It employs a decoder to reconstruct images from encoded features. This allows the iden-

*Corresponding author.

tification of input data containing a trigger. However, a notable limitation of this approach is the need to train a decoder on an auxiliary dataset, which may present practical challenges in real-world scenarios. Overall, current methodologies primarily focus on detecting and eliminating backdoors in CL models that have already been trained and compromised. However, this reactive approach suffers from several fundamental limitations. Once a model is done training, the damage is largely done, making effective remediation challenging. Eliminating an integrated backdoor often necessitates either expensive re-training or aggressive weight-pruning that could degrade the model's utility. Also, the lack of effective and tailored filtering algorithm lead to large ASR residual after re-training. Furthermore, preventing backdoor activation during inference time (Hou, Li, and Yao 2024) typically requires the deployment of an auxiliary decoder, adding further complexity.

In contrast to these reactive approaches, our research advocates for a proactive strategy: defending against backdoors during the training process. We propose Density-based Identification and Fine-tuning (DIFT), a two-stage defense framework specifically designed to safeguard CL against backdoor attacks. DIFT addresses backdoor threats during training by employing a sophisticated detection pipeline that identifies and filters out poisoned data during the early training phase. The first stage of DIFT fast searches for poisoned embeddings with abnormally high density during training using their local distance to neighbors as an indicator. It further sanitizes the identified embeddings by their distances to anchor samples, which allows us to remove false positives and achieve accurate detection. Subsequently, to enhance the coverage of poisoned data, semi-supervised label spreading is employed to pinpoint all similar points. The second stage of DIFT focuses on reducing residual backdoor effects. Instead of retraining the model, we employ a fine-tuning approach using the detected set of poisoned samples. This is achieved by eliminating the distinctive characteristics of poisoned embeddings. Finally, training on the clean training dataset can be proceeded. This in-training approach offers high-level advantage that allows us to leverage control over the training dynamics. Early detection of poisoning samples enables DIFT to apply mitigation at early stage, resulting in improved performance and robustness.

Overall, our work explores the promising direction of high-density property of poisoned embeddings. We propose a novel method that integrates defense directly into the model's learning process for a more principled and effective resolution. **Our contributions** are:

- We propose DIFT, a novel in-training defense framework that mitigates the backdoor in CL. Our scheme can avoid the problematic backdoor removal post-training by detecting poisoned samples and eliminate backdoor effect in early stage of training.
- We propose a sophisticated identification pipeline that exploits the high-density clustering property of poisoned samples. It refines and expands the poisoning candidate to accurately detect all malicious data. Moreover, we propose a fine-tuning stage that targets the poisoning characteristic to remove the residual effects of backdoors.
- Through extensive experiments, we demonstrate that our defense can successfully resist various backdoor attacks against CL. After applying DIFT, the backdoor success rate drops significantly, while the accuracy on clean data is maintained.

Related Work

Contrastive Learning

InstDisc (Wu et al. 2018) first introduced the concept of instance discrimination, using a memory bank to store features. It treats each data sample as an independent instance, and the model learns to map these instances into a unique low-dimensional space using a memory bank to store features. InstDisc lays the groundwork for future methods. MoCo (He et al. 2020) improves representation learning by using a dynamic dictionary and a momentum-based moving average encoder. This method followed the instance discrimination paradigm but enhanced scalability and performance. SimCLR (Chen et al. 2020) simplifies the CL framework by eliminating the need for a memory bank and introducing a learnable nonlinear transformation layer between the representation and the contrastive loss. BYOL (Grill et al. 2020) removes the need for negative samples by reformulating the problem as a prediction task, where one augmented view of an image is used to predict the representation of another augmented view. SimSiam (Chen and He 2021) employs an Expectation-Maximization-like algorithm by splitting parameters into separate prediction and target networks, demonstrating strong empirical performance. CL has emerged as a critical component of AI research and applications. However, its widespread adoption also makes it a significant target for various attacks, which could lead to extensive and far-reaching consequences.

Data Poisoning Backdoor Attacks

The popularity of CL also spurs intensive research on its security properties (Li et al. 2024; Saha et al. 2022). Backdoor attacks pose a significant security threat to it. A common approach to implanting backdoors is data poisoning. The attacker injects poisoned samples containing predefined triggers into the training data. The CL training on poisoned data is defined as:

$$\mathcal{L}_{\text{total}} = \underbrace{\mathbb{E}_{\substack{z\sim\mathcal{D} \\ z^+\sim\mathcal{A}(z)}}[\ell_\theta(z, z^+)]}_{\text{Standard Contrastive Loss}} + \underbrace{\mathbb{E}_{\substack{z_p\sim\mathcal{P} \\ z_p^+\sim\mathcal{A}(z_p)}}[\ell_\theta^{\text{poison}}(z_p, z_p^+)]}_{\text{Backdoor Injection Term}} \tag{1}$$

where $\mathcal{D}$ is the clean dataset, $\mathcal{P}$ is the poisoned dataset, and v is the input data. (Zhang et al. 2024a) successfully poisoned the model through exploiting the random cropping mechanism, a common data augmentation technique used in CL. BadEncoder (Jia, Liu, and Gong 2022) injects a backdoor into pre-trained clean encoders and releases the model to compromise downstream tasks. Instead of poisoning the training data, the authors of this work fine-tuned the model with knowledge of the downstream task. SSLBackdoor (Saha et al. 2022) uses a randomly positioned fixed

patch (*e.g.*, 5×5) as a backdoor trigger to poison the target model. PoisonedEncoder (Liu, Jia, and Gong 2022) generates poisoned samples by randomly combining target inputs with reference inputs. CTRL (Li et al. 2023) is among the most effective backdoor poisoning attacks. It embeds a trigger by increasing the magnitude of a specific frequency component. This approach ensures both stealthiness and effectiveness.

Threat Model

Objective. The attacker intends to stealthily inject a backdoor into the target CL model so that when the model is used as an encoder for a downstream task, the embeddings become misleading, causing the downstream classifier to make incorrect predictions. The attacker also need to ensure that the downstream classifier performs well on features encoded from clean data without the trigger.

Attacker's ability and capacity. We assume that the attacker can only manage to pollute a small percentage of poisoned samples into the dataset used for encoder training. The attacker has no control over the training process and no knowledge of it, including the training algorithm, hyperparameters, model architecture, and optimizer. To perform this attack, the attacker collects samples exclusively from the target class, which is the class into which inputs are desired to be misclassified into, and embeds triggers into them. The poisoned data is then released publicly, making it accessible for collection by encoder trainers, who may unknowingly incorporate it into their training datasets. The attacker also has no control over the training of the downstream classifier. It cannot manipulate the training process, nor can it pollute any of the classifier's training data.

Methodology

Overview

As illustrated in Figure 1, DIFT consists of two main stages: 1) **Detection of Poisoned Samples:** We identify poisoned training samples using a sophisticated density-based filtering pipeline. To reduce false positives, we apply a filtering step and intersect the results from those of different checkpoints. Label spreading further propagates the detected poisoned samples from high-density regions to improve detection coverage. 2) **Mitigation of Poisoned Samples' Influence:** After identifying the poisoned samples, we fine-tune the model to mitigate their impact. To achieve this, we introduce a custom loss function designed to reduce the density of poisoned samples in the feature space. Specifically, we train the model on the detected poisoned samples to increase the average pairwise distance between them. After this, we continue training on the filtered dataset while restarting the learning rate scheduler.

Poisoned Sample Identification

It is observed in our experiments and also demonstrated in previous work (Li et al. 2024), the embeddings of poisoned samples tend to cluster together with low pair-wise L_2 distance. We argue that this is an inherent behavior of model with backdoor. Here, we provide analysis toward this claim. For the convenience of the analysis, we make the following assumptions.

Assumption 1. Assume that the norm of the input z is upper bounded by $\frac{D}{2}$.

Remark. The assumption 1 is common in the task of learning image representation, since each input belongs to $[0,1]^d$.

Assumption 2. Assume that the loss function $\mathcal{L}$ is L_Θ-lipschitz on the hypothesis space and input space, where $\frac{\|\mathcal{L}(z,\theta)-\mathcal{L}(z,\theta^{'})\|}{\|\theta-\theta^{'}\|} \leq L_\Theta$ for any $z \in \mathcal{Z}$, and $\frac{\|\mathcal{L}(z,\theta)-\mathcal{L}(z^{'},\theta)\|}{\|z-z^{'}\|} \leq L_\mathcal{Z}$ for any $\theta \in \Theta$.

Assumption 3. Assume that the loss function $\mathcal{L}$ is s_Θ-smooth to the hypothesis space.

Theorem 4. *Given encoder f with parameter θ_0, iteration $K \geq 2$, learning rate η, loss function $\mathcal{L}$, poisoned training data set $Z = \{z_1, ..., z_n\}$, and further assume the poison ratio $\gamma > 0$ and the backdoor strength $C \geq 1$ of the backdoor algorithm $\mathcal{A}$, where $C\|\mathcal{A}(z) - \mathcal{A}(z^{'})\| \leq \|z - z^{'}\|$, omitting some constants, we have*

$$\mathbb{E}_\mathcal{A}\left[\|f(z,\theta_K) - f(z^{'},\theta_K)\|\right] = \mathcal{O}(\frac{1}{C}, \frac{1}{\gamma^2}, D) \quad (2)$$

Proof. According to the Triangle Inequality, for $0 \leq i \leq K-1$,

$$\begin{aligned}
&\mathbb{E}_\mathcal{A}\left[\|f(z,\theta_{i+1}) - f(z^{'},\theta_{i+1})\| - \|f(z,\theta_i) - f(z^{'},\theta_i)\|\right] \\
\leq &\mathbb{E}_\mathcal{A}\left[\|f(z,\theta_{i+1}) - f(z,\theta_i) + f(z^{'},\theta_{i+1}) - f(z^{'},\theta_i)\|\right] \\
\leq &\mathbb{E}_\mathcal{A}\left[\|f(z,\theta_{i+1}) - f(z,\theta_i)\| + \|f(z^{'},\theta_{i+1}) - f(z^{'},\theta_i)\|\right] \\
\leq &2L_\Theta \mathbb{E}_\mathcal{A}\left[\|\theta_{i+1} - \theta_i\|\right] \\
= &\frac{2L_\Theta}{B}\mathbb{E}_\mathcal{A}\left[\|\Sigma_{j=1}^B \nabla_\theta \mathcal{L}(\theta_i, z_j)\|\right]
\end{aligned}$$

Now consider the average norm of the gradient from a batch in the i-th iteration,

$$\begin{aligned}
&\mathbb{E}_\mathcal{A}\left[\frac{1}{B}L_\Theta\|\Sigma_{m=1}^B \nabla_\theta \mathcal{L}(\theta_{i+1}, z_{p_m})\| - \right. \\
&\qquad\qquad \left. \frac{1}{B}L_\Theta\|\Sigma_{m=1}^B \nabla_\theta \mathcal{L}(\theta_i, z_{q_m})\|\right] \\
\leq &\frac{1}{B}\mathbb{E}_\mathcal{A}\left[\|\Sigma_{j=1}^B \nabla_\theta \mathcal{L}(\theta_{i+1}, z_{p_m}) - \Sigma_{m=1}^B \nabla_\theta \mathcal{L}(\theta_i, z_{q_m})\|\right] \\
\leq &\frac{1}{B}\mathbb{E}_\mathcal{A}\left[\|\Sigma_{j=1}^B \nabla_\theta \mathcal{L}(\theta_{i+1}, z_{p_m}) - \Sigma_{m=1}^B \nabla_\theta \mathcal{L}(\theta_i, z_{p_m})\|\right] \\
&+ \frac{1}{B}\mathbb{E}_\mathcal{A}\left[\|\Sigma_{m=1}^B \nabla_\theta \mathcal{L}(\theta_i, z_{p_m}) - \Sigma_{m=1}^B \nabla_\theta \mathcal{L}(\theta_i, z_{q_m})\|\right] \\
\leq &s_\Theta + \frac{1}{B}\mathbb{E}_\mathcal{A}\left[\Sigma_{m=1}^B \|z_{p_m} - z_{q_m}\|\right] \\
\leq &s_\Theta + \frac{1}{B}\left((1-\gamma^2)D + \gamma^2\frac{D}{C}\right) \\
= &s_\Theta + \frac{\left(C + (1-C)\gamma^2\right)D}{BC}
\end{aligned}$$

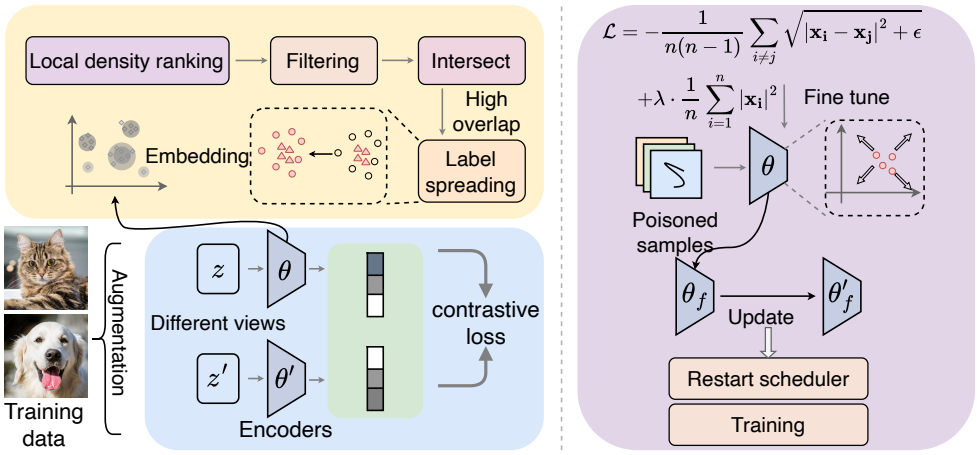


Figure 1: The workflow of our proposed *DIFT*. The encoded features of the training data are used for poisoned sample identification. Our detection method consists of four phases: local density ranking, filtering, intersection with previous results, and label spreading to detect and include all poisoned samples. Subsequently, we fine-tune the encoder on poisoned data to eliminate the backdoor effect using a loss function designed to reduce density. Finally, we restart the learning rate scheduler and complete the remaining training epochs.

Add the norm of the gradient of each batch recursively, then,

$$\mathbb{E}_{\mathcal{A}}\left[\Sigma_{j=1}^{B}\nabla_{\theta}\mathcal{L}(\theta_i, z_j)\right] \leq (K-2)\left(s_{\Theta}+\frac{\left(C+(1-C)\gamma^2\right)D}{BC}\right)+L_{\Theta}$$

Therefore, by adding $\mathbb{E}_{\mathcal{A}}\left[\Sigma_{j=1}^{B}\nabla_{\theta}\mathcal{L}(\theta_i, z_j)\right]$ for each $0 \leq i \leq K-1$, we have

$$\mathbb{E}_{\mathcal{A}}\left[\|f(z,\theta_K)-f(z^{'},\theta_K)\|\right] \leq \frac{2L_{\Theta}(K-1)}{B}\left((K-2)\left(s_{\Theta}+\frac{\left(C+(1-C)\gamma^2\right)D}{BC}\right)+L_{\Theta}\right)$$

□

Theorem 4 shows that there exists an upper bound of the expected distance between representations of any two inputs are negatively related with the contraction coefficient C, poison ratio γ, and positively related with the maximum input norm $\frac{D}{2}$. Therefore, if the poison ratio γ and the backdoor strength C are large enough, representations of any two inputs will be nearly identical, indicates that backdoor triggers are the only learnable feature in this case.

Proposition 5. *Assume there exists an another contraction coefficient $C^{'} > 1$ of the backdoor attack $\mathcal{A}$ in the embedding space, so that $C^{'}\|f(\mathcal{A}(z)) - f(\mathcal{A}(z^{'}))\| \leq \|f(z) - f(z^{'})\|$, we have,*

$$\mathbb{E}_{\mathcal{A}}\left[\|f(\mathcal{A}(z),\theta_K)-f(\mathcal{A}(z^{'}),\theta_K)\|\right] = \mathcal{O}(\frac{1}{C},\frac{1}{C'},\frac{1}{\gamma^2},D) \tag{3}$$

Following the proof of Theorem 4, Proposition 5 could be proved similarly, which upper bounds the distance between the representations of any two backdoored sample, showing the clustering phenomenon aligned with the observation. We also provide a simple analysis from the information theoretical point of view, which can be found in Appendix.

During training, we examine the embeddings of training data to identify poisoned data after each epoch. To be specific, we first compute the k-nearest neighbors (KNN) distance for the embeddings as

$$d_k(x_i) = \min\left\{r \mid \sum_{x_j \in X\setminus\{x_i\}} \mathbb{I}\left(d(x_i,x_j) \leq r\right) \geq k\right\} \tag{4}$$

where x is the encoded embeddings. The local density of a point can be seen as the inverse of the distance to its k^{th} nearest neighbor:

$$\rho_i = \frac{1}{d_k(x_i)} \tag{5}$$

We use the Hierarchical Navigable Small World (HNSW) algorithm (Malkov and Yashunin 2018) for efficient KNN searches. The points with the highest density, that is, those with the smallest k^{th} nearest neighbor distance, are selected as poisoned candidates. However, since the proportion of poisoned data in the attacks is usually very low, some normal data points may also exhibit high local density. To address this, we use anchor points a to differentiate poisoned data from normal data. We assume access to a small subset of clean training data, consistent with prior work (Yue et al. 2023; Zeng et al. 2022; Li et al. 2021b). Unlike previous approaches, we require only ten anchor samples per class in the clean subset, which represents a significantly lower requirement compared to the commonly used 5% clean data

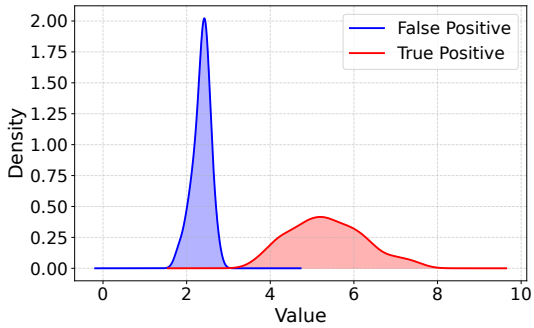


Figure 2: KDE plot of MinDist of high-density points.

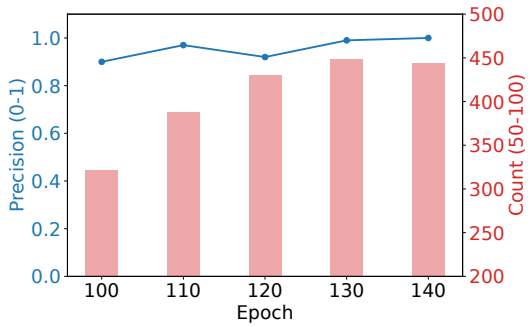


Figure 3: Precision and detected sample number before label spreading.

setting (Yue et al. 2023; Li et al. 2021b). We compute the average distance between a poisoned candidate x and the anchor points of each class. We then calculate MinDist(x) to the class with the smallest average distance as

$$\text{MinDist}(x) = \min_{i \in \{1,2,\ldots,c\}} \left(\frac{1}{n} \sum_{j=1}^{n} \|x - a_{i,j}\| \right) \quad (6)$$

The intuition is that every benign embedding has anchor points from a nearby class close to it. To demonstrate the difference, we perform kernel density estimation (KDE) on the calculated distances from backdoored model. Using SimCLR (Chen et al. 2020) on CIFAR10 as an example, the results are shown in Figure 2. We select the highest-density points from the encoded features. The embeddings contain a mix of true positives (poisoned samples) and false positives (benign samples). As shown in the figure, the distance values of poisoned and benign samples are well separated. To facilitate efficiently and automatically separation, we employ a Gaussian Mixture Model (GMM) to separate these points and filter out samples with high distance values. During training, the samples filtered in this step are stored in a buffer. Each new result is intersected with the previous one until the set of filtered samples stabilizes. Our algorithm converges quickly, enabling the detection of poisoned samples in the early stages of training. Figure 3 shows the precision and the final number of detected samples during training. It shows that the detected sample counts stabilizes in early epochs. The pseudocode of this algorithm is presented in Appendix.

The objective of aforementioned operations is to identify indicative poisoned samples from the training data. However, these steps may fail to detect all poisoned instances, which will lead to incomplete detection. To address this limitation, we introduce an expansion strategy that builds upon the initially identified samples to recover all poisoned data. To achieve this, we apply the semi-supervised Label Spreading algorithm (Zhou et al. 2003). We assign a label of 1 to the current poisoned candidates, -1 to the anchor samples, and 0 to all other samples. All samples assigned a label of 1 after applying label spreading form the final set of identified poisoned samples.

Fine-Tuning for Density Reduction

Retraining the model with clean data after identifying poisoned samples is the standard approach for model sanitization (Li et al. 2024). However, this approach is ineffective and computationally expensive. In this work, we hope to mitigate the backdoor effect during training after its discovery and continue training without full model retraining. Therefore, we propose to fine-tune the encoder using the identified poisoned samples.

To eliminate the backdoor, fine-tuning aims to remove the key characteristic of the backdoor, which is high local density. Therefore, we fine-tune the encoder on the identified poisoned samples by optimizing the following loss function:

$$\mathcal{L}_{ft} = \underbrace{-\frac{1}{n(n-1)} \sum_{i \neq j} \sqrt{|\mathbf{x_i} - \mathbf{x_j}|^2 + \epsilon}}_{\text{reduce density}} + \underbrace{\lambda \cdot \frac{1}{n} \sum_{i=1}^{n} |\mathbf{x_i}|^2}_{\text{regularizer}} \quad (7)$$

where the first term increases the mean pairwise distance between feature vectors, and the second term prevents their magnitudes from growing infinity. ϵ is a small constant for numerical stability, and λ is the regularization constant. The small number of poisoned samples makes our fine-tuning process efficient and incurs minimal computational overhead. We illustrate the effect of training with poisoned data in Figure 4. The attacker manipulates the global model by shifting its parameters from the benign optimum toward a backdoored optimum. Fine-tuning is necessary because, even after the backdoor is embedded, the model retains non-trivial performance on clean data. This residual utility leads to small update steps during training, which may cause the model to converge to a local minimum aligned with the backdoored objective. Through fine-tuning, we guide the model out of this backdoor-induced local minimum and promote recovery of its clean predictive behavior. It is worth noting that many CL algorithms maintain two branches during training. For instance, BYOL (Grill et al. 2020) maintains an online and a target branch, while MoCo (He et al. 2020) maintains a key encoder using a momentum-based moving average of the query encoder. During fine-tuning, we optimize only the encoder θ branch, which is used for encoding after training. This results in asymmetry between

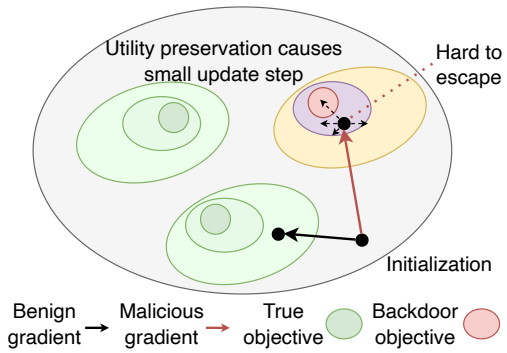


Figure 4: Model weights update under backdoor attack.

the two branches. Therefore, we manually update the parameters of the other branch θ' after fine-tuning. Specifically, we copy the exact fine-tuned parameters θ_{ft} to the other branch. When copying parameters, we exclude the predictor from the online branch in BYOL and reset the queue in MoCo. The pseudocode of this is shown in Appendix. After this step, we restart the training process. We reset the learning rate scheduler and complete the remaining training epochs. This involves reapplying the scheduler from the initial learning rate with fewer remaining epochs.

Experiments

Experimental Setup

Dataset	Algorithm	Acc (%)	ASR (%)
CIFAR10	SimCLR	87.4	99.0
	MoCo	82.5	97.8
	BYOL	83.4	98.4
CIFAR100	SimCLR	49.4	96.2
	MoCo	46.1	95.9
	BYOL	48.9	96.0
ImageNet-100	SimCLR	50.2	49.1
	MoCo	48.8	67.1
	BYOL	49.0	53.8

Table 1: Baseline accuracy (%) and ASR (%) of algorithms on benign samples without defense

We first introduce the setup of our experimental evaluation. Additional details are provided in the Appendix.

Implementation Settings We evaluate three CL algorithms: SimCLR (Chen et al. 2020), MoCo (He et al. 2020), and BYOL (Grill et al. 2020), using three benchmark datasets: CIFAR-10, CIFAR-100 (Krizhevsky 2009), and ImageNet-100 (Le and Yang 2015). ImageNet-100 is a subset that randomly sampled from ImageNet-1K dataset (Deng et al. 2009) and rescaled down to 64×64. We primarily use three backdoor poisoning attacks, namely frequency-based CTRL (Li et al. 2023), patch-based SSL-Backdoor (Saha et al. 2022), and CorruptEncoder (Zhang et al. 2024a). Our implementation runs on a Nvidia H100 GPU (80G). The implementation detail and training hyperparameters are shown in Appendix.

Attack Settings The perturbation frequency of (Li et al. 2023) is fixed at 15 and 31 and the perturbation magnitude is 100. A fixed-size (e.g., 5×5) applied to a pre-defined position is defined as the trigger for (Saha et al. 2022). Poisoning samples from (Zhang et al. 2024a) are generated using the reference and the background images, as specified in the original paper. We fix the poisoning ratio to 1% across all settings.

Selection of Baselines We adopt two defense baselines proposed in (Li et al. 2024) namely density-based filtering followed by re-training and ensemble defense combining filtering and data-free pruning. The adopted density-based clustering algorithm is DBSCAN.

Metrics We evaluate our method using two primary metrics, namely accuracy (ACC) and ASR. We aim to achieve a significant reduction in ASR while ensuring minimal impact on ACC. We use precision and recall as performance metrics to evaluate the poisoned sample identification in ablation study.

Results

Defense effectiveness To defend against the backdoor attack, we evaluate and compare the effectiveness of our approach with baselines. Table 1 list the performance of CTRL on the target datasets. We use these metrics as benchmark for following comparison. We report the defense results of (Li et al. 2023) in Table 2. Additional results of (Saha et al. 2022) and (Zhang et al. 2024a) are provided in the Appendix. Based on our experiments, we observe the following phenomena: *(i)* The integration of DIFT demonstrates a significant reduction in ASR across all experimental settings. *(ii)* The integration of DIFT into the training process slightly reduces the model's performance, as reflected in accuracy. This indicates that despite a short interruption during training, DIFT preserves the utility of the encoder. *(iii)* DIFT's effectiveness generalizes on various backdoor attacks. *(iv)* Despite the efficacy of DIFT in mitigating adversarial influences, a residual ASR persists at non-zero levels. Excluding the possibility of misclassification, we think that this residual ASR indicates minimal backdoor efficacy. However, this is an acceptable security-performance trade-off within practical constraints.

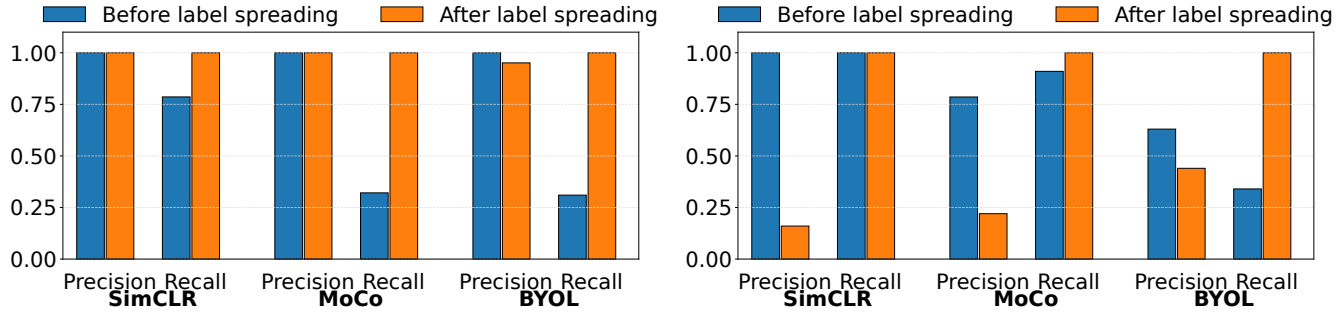
We have also provided comparisons of DIFT with defense baselines in terms of raw filtering performance. For clustering-based detection methods such as DBSCAN and OPTICS (Li et al. 2024), we report the TPR and FPR in comparison to our results. In addition, we compare our method with DeDe (Hou, Li, and Yao 2024) by reporting the time consumption to demonstrate superiority. Please refer to the detailed results provided in the Appendix.

Ablation study The necessity of this multi-stage design is confirmed by our comprehensive ablation study.

The detection effectiveness before label spreading is well discussed in the previous section. We focus on evaluating the effectiveness of label spreading. The effectiveness of

Dataset	Algorithm	Acc w/ Defense (%)			Acc Drop (%) ↓			ASR w/ Defense (%)			ASR Drop (%) ↓		
		B1	B2	DIFT	B1	B2	DIFT	B1	B2	DIFT	B1	B2	DIFT
CIFAR-10	SimCLR	87.0	86.0	**87.0**	0.4	1.4	**0.4**	47.4	13.1	**2.0**	51.6	85.9	**97.0**
	MoCo	80.4	78.3	**80.5**	2.1	4.2	**2.0**	51.9	14.6	**2.5**	45.9	83.2	**95.3**
	BYOL	80.1	78.4	**83.2**	3.3	5.0	**0.1**	52.4	27.7	**1.5**	46.0	70.7	**96.9**
CIFAR-100	SimCLR	47.9	46.1	**48.1**	1.5	3.3	**1.3**	74.7	35.3	**0.7**	21.5	60.9	**95.5**
	MoCo	44.0	42.1	**45.4**	2.1	4.0	**0.7**	65.6	29.8	**0.0**	30.3	66.1	**95.9**
	BYOL	47.8	45.9	**47.9**	1.1	3.0	**1.0**	63.0	33.4	**0.0**	33.0	62.6	**96.0**
ImageNet-100	SimCLR	**46.0**	45.1	45.9	**4.2**	5.1	4.3	37.1	19.2	**5.3**	12.0	29.9	**43.8**
	MoCo	45.1	42.5	**45.5**	3.7	6.3	**3.3**	48.8	27.7	**5.2**	18.3	39.4	**61.9**
	BYOL	44.5	42.8	**44.6**	4.5	6.2	**4.4**	41.6	22.3	**3.6**	12.2	31.5	**50.2**

Table 2: Performance comparison of defense schemes. B1 is the density-based filtering followed by re-training, and B2 is the ensemble method combining re-training and data-free pruning.



(a) CIFAR10 (b) CIFAR100

Figure 5: Ablation study of label spreading. We plot the precision and recall scores for comparison. For the implementation and configuration of label spreading, we use the default settings from (Pedregosa et al. 2011) with an RBF kernel.

DIFT lies in its ability to detect poisoned samples both accurately and comprehensively. Therefore, achieving high recall in poisoned data identification is critical for the success of DIFT. However, identifying all poisoned samples remains challenging due to two primary limitations. First, preceding steps prioritize minimizing false positives, which can accidentally exclude some poisoned samples. Second, per-sample high-density filtering may fail to identify poisoned samples near the edges of clusters or may be misled by benign samples that occasionally exhibit higher local density. Label spreading addresses these issues by incorporating samples similar to those initially identified.

Figure 5 presents the precision and recall before and after label spreading. Low recall before and high recall after demonstrate the effectiveness of label spreading of capturing all poisoned samples. While in some cases the precision decreases afterwards, the number of false positives remains small compared to the size of the training set. Therefore, we argue that this error is within an acceptable range, as it is caused by the low poisoning rate, i.e., a low number of true positives. The low Acc drop suggests that despite some false positive filtering, the model's utility remains well-preserved as shown in Table 2. We also report ablation study on the final model performance, namely ASR and Acc, which is deferred to the Appendix. The finding is that the model trained without label spreading exhibit high ASR residual.

The ablation study on fine-tuning is deferred to the Appendix. However, the findings are consistent across settings, showing that fine-tuning is critical for defense effectiveness. For instance, on CIFAR-10 with SimCLR, removing the fine-tuning step results in a residual ASR of 86.4%.

Conclusion

In this paper, we address the critical security challenge of backdoor attacks in CL. We propose DIFT, a novel and effective defense framework to mitigate this threat during training. By leveraging the inherent density distinction of poisoned samples in the feature space, DIFT utilize an effective detection pipeline to identify poisoned samples. Furthermore, in order to mitigate the backdoor effect while preserving the model's performance on clean data, we fine-tune the encoder with an objective to reduce the density of poisoned samples. Empirical evaluations across multiple CL algorithms and datasets demonstrate the effectiveness of DIFT. Future work includes exploring the applicability of our density-based defense strategy to other self-supervised learning paradigms and developing adaptive defenses against backdoor attacks implanted after training (Tao et al. 2024; Jia, Liu, and Gong 2022).

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No: 92270123 and 62372122), Heilongjiang Key R&D Program of China (Grant No. GA23A915), National Key Support Program for Foreign Experts (Grant No. D20240249), and the Research Grants Council (Grant No: 15209922, 15208923), Hong Kong SAR, China.

References

Adi, Y.; Baum, C.; Cisse, M.; Pinkas, B.; and Keshet, J. 2018. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In *27th USENIX security symposium (USENIX Security 18)*, 1615–1631.

Ankerst, M.; Breunig, M. M.; Kriegel, H.-P.; and Sander, J. 1999. OPTICS: Ordering points to identify the clustering structure. *ACM Sigmod record*, 28(2): 49–60.

Bagdasaryan, E.; and Shmatikov, V. 2021. Blind backdoors in deep learning models. In *30th USENIX Security Symposium (USENIX Security 21)*, 1505–1521.

Bai, L.; Hu, H.; Ye, Q.; Xu, J.; Li, J.; Fang, C.; and Shi, J. 2025. RMR: A Relative Membership Risk Measure for Machine Learning Models. *IEEE Transactions on Dependable and Secure Computing*, 22(5): 4699–4710.

Bengio, Y.; Courville, A.; and Vincent, P. 2013. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8): 1798–1828.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, 1597–1607. PmLR.

Chen, X.; and He, K. 2021. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 15750–15758.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Ester, M.; Kriegel, H.-P.; Sander, J.; Xu, X.; et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, 226–231.

Feng, S.; Tao, G.; Cheng, S.; Shen, G.; Xu, X.; Liu, Y.; Zhang, K.; Ma, S.; and Zhang, X. 2023. Detecting backdoors in pre-trained encoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16352–16362.

Gao, Y.; Xu, C.; Wang, D.; Chen, S.; Ranasinghe, D. C.; and Nepal, S. 2019. Strip: A defence against trojan attacks on deep neural networks. In *Proceedings of the 35th annual computer security applications conference*, 113–125.

Goodfellow, I.; Bengio, Y.; and Courville, A. 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

Grill, J.-B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P.; Buchatskaya, E.; Doersch, C.; Avila Pires, B.; Guo, Z.; Gheshlaghi Azar, M.; et al. 2020. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33: 21271–21284.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9729–9738.

Hou, S.; Li, S.; and Yao, D. 2024. DeDe: Detecting Backdoor Samples for SSL Encoders via Decoders. *arXiv preprint arXiv:2411.16154*.

Jia, J.; Liu, Y.; and Gong, N. Z. 2022. Badencoder: Backdoor attacks to pre-trained encoders in self-supervised learning. In *2022 IEEE Symposium on Security and Privacy (SP)*, 2043–2059. IEEE.

Krizhevsky, A. 2009. Learning Multiple Layers of Features from Tiny Images. Technical report, University of Toronto.

Le, Y.; and Yang, X. 2015. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7): 3.

Li, C.; Pang, R.; Cao, B.; Xi, Z.; Chen, J.; Ji, S.; and Wang, T. 2024. On the difficulty of defending contrastive learning against backdoor attacks. In *33rd USENIX Security Symposium (USENIX Security 24)*, 2901–2918.

Li, C.; Pang, R.; Xi, Z.; Du, T.; Ji, S.; Yao, Y.; and Wang, T. 2023. An embarrassingly simple backdoor attack on self-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4367–4378.

Li, H.; Bai, L.; Ye, Q.; Hu, H.; Xiao, Y.; Zheng, H.; and Xu, J. 2025. A Sample-Level Evaluation and Generative Framework for Model Inversion Attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 18287–18295.

Li, Y.; Li, Y.; Wu, B.; Li, L.; He, R.; and Lyu, S. 2021a. Invisible backdoor attack with sample-specific triggers. In *Proceedings of the IEEE/CVF international conference on computer vision*, 16463–16472.

Li, Y.; Lyu, X.; Koren, N.; Lyu, L.; Li, B.; and Ma, X. 2021b. Neural Attention Distillation: Erasing Backdoor Triggers from Deep Neural Networks. In *International Conference on Learning Representations*.

Liang, S.; Zhu, M.; Liu, A.; Wu, B.; Cao, X.; and Chang, E.-C. 2024. Badclip: Dual-embedding guided backdoor attack on multimodal contrastive learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 24645–24654.

Liu, H.; Jia, J.; and Gong, N. Z. 2022. {PoisonedEncoder}: Poisoning the unlabeled pre-training data in contrastive learning. In *31st USENIX Security Symposium (USENIX Security 22)*, 3629–3645.

Malkov, Y. A.; and Yashunin, D. A. 2018. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence*, 42(4): 824–836.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss,

R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Duchesnay, E. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830.

Saha, A.; Tejankar, A.; Koohpayegani, S. A.; and Pirsiavash, H. 2022. Backdoor Attacks on Self-Supervised Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 13337–13346.

Tang, L.; Ye, Q.; Hu, H.; Xue, Q.; Xiao, Y.; and Li, J. 2024. DeepMark: A Scalable and Robust Framework for DeepFake Video Detection. *ACM Trans. Priv. Secur.*, 27(1).

Tao, G.; Wang, Z.; Feng, S.; Shen, G.; Ma, S.; and Zhang, X. 2024. Distribution preserving backdoor attack in self-supervised learning. In *2024 IEEE Symposium on Security and Privacy (SP)*, 2029–2047. IEEE.

Wang, H.; Guo, S.; He, J.; Chen, K.; Zhang, S.; Zhang, T.; and Xiang, T. 2024. Eviledit: Backdooring text-to-image diffusion models in one second. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 3657–3665.

Wang, H.; Guo, S.; He, J.; Liu, H.; Zhang, T.; and Xiang, T. 2025. Model Supply Chain Poisoning: Backdooring Pre-trained Models via Embedding Indistinguishability. In *Proceedings of the ACM on Web Conference 2025*, 840–851.

Wang, Y.; Zhu, Y.; and Gao, X.-S. 2024. Efficient availability attacks against supervised and contrastive learning simultaneously. *Advances in Neural Information Processing Systems*, 37: 72872–72900.

Wu, Z.; Xiong, Y.; Yu, S. X.; and Lin, D. 2018. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3733–3742.

Xiao, Y.; Hu, H.; Ye, Q.; Tang, L.; Liang, Z.; and Zheng, H. 2025. Unlocking High-Fidelity Learning: Towards Neuron-Grained Model Extraction. *IEEE Transactions on Dependable and Secure Computing*.

Yue, Z.; Xia, J.; Ling, Z.; Hu, M.; Wang, T.; Wei, X.; and Chen, M. 2023. Model-contrastive learning for backdoor elimination. In *Proceedings of the 31st ACM International Conference on Multimedia*, 8869–8880.

Zeng, Y.; Chen, S.; Park, W.; Mao, Z.; Jin, M.; and Jia, R. 2022. Adversarial Unlearning of Backdoors via Implicit Hypergradient. In *International Conference on Learning Representations*.

Zhang, J.; Liu, H.; Jia, J.; and Gong, N. Z. 2024a. CorruptEncoder: Data Poisoning based Backdoor Attacks to Contrastive Learning.

Zhang, J.; Liu, H.; Jia, J.; and Gong, N. Z. 2024b. Data poisoning based backdoor attacks to contrastive learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 24357–24366.

Zhang, X.; Hu, H.; Ye, Q.; Bai, L.; and Zheng, H. 2025. Merinspector: Assessing model extraction risks from an attack-agnostic perspective. In *Proceedings of the ACM on Web Conference 2025*, 4300–4315.

Zheng, M.; Xue, J.; Wang, Z.; Chen, X.; Lou, Q.; Jiang, L.; and Wang, X. 2024. Ssl-cleanse: Trojan detection and mitigation in self-supervised learning. In *European Conference on Computer Vision*, 405–421. Springer.

Zheng, R.; Tang, R.; Li, J.; and Liu, L. 2022. Data-free backdoor removal based on channel lipschitzness. In *European Conference on Computer Vision*, 175–191. Springer.

Zhou, D.; Bousquet, O.; Lal, T.; Weston, J.; and Schölkopf, B. 2003. Learning with local and global consistency. *Advances in neural information processing systems*, 16.