Leveraging Imperfect Restoration for Data Availability Attack

Yi Huang[1*], Jeremy Styborski[1*], Mingzhi Lyu[2*], Fan Wang[2], and Adams Kong[1]

[1] College of Computing and Data Science, Nanyang Technological University, Singapore
[2] Rapid-Rich Object Search (ROSE) Lab, Interdisciplinary Graduate Programme, Nanyang Technological University, Singapore
shellbyhuang@gmail.com, {styb0001,lyum0002,fan005,AdamsKong}@ntu.edu.sg

Abstract. The abundance of online data is at risk of unauthorized usage in training deep learning models. To counter this, various Data Availability Attacks (DAAs) have been devised to make data unlearnable for such models by subtly perturbing the training data. However, existing attacks often excel against either Supervised Learning (SL) or Self-Supervised Learning (SSL) scenarios. Among these, a model-free approach that generates a Convolution-based Unlearnable Dataset (CUDA) stands out as the most robust DAA across both SSL and SL. Nonetheless, CUDA's effectiveness against SSL is underwhelming and it faces a severe trade-off between image quality and its poisoning effect. In this paper, we conduct a theoretical analysis of CUDA, uncovering the sub-optimal gradients it introduces and elucidating the strategy it employs to induce class-wise bias for data poisoning. Building on this, we propose a novel poisoning method named Imperfect Restoration Poisoning (IRP), aiming to preserve high image quality while achieving strong poisoning effects. Through extensive comparisons of IRP with eight baselines across SL and SSL, coupled with evaluations alongside five representative defense methods, we showcase the superiority of IRP. **Code:** `https://github.com/lyumingzhi/IRP`

Keywords: Data Availability Attacks · Supervised Learning · Self-Supervised Learning

1 Introduction

The proliferation of online data has proven an indispensable resource for the advancement of deep learning models. However, the collection of certain datasets without explicit consent poses a possible threat to personal privacy [2]. Additionally, the emergence of generative AI presents new challenges, as training with unauthorized data could infringe upon the owners' copyright [34].

DAAs have emerged as a promising strategy to address the issue of unauthorized data usage [12–14, 17, 20, 27, 29, 33, 40–42]. These attacks introduce subtle

[*] Equal contribution

perturbations into training data, hindering the model's ability to effectively learn useful information, which subsequently leads to poor performance on unseen data. This poor performance manifests as a substantial decrease in accuracy when the model is tested on clean data in classification tasks. Currently, the majority of DAAs are developed within the framework of SL. However, a recent study by He et al. [17] reveals that Adversarial Poison (AP) [13], a cutting-edge DAA targeting SL, lacks effectiveness when applied to SSL. In response, they introduce a novel approach, termed Contrastive Poison (CP) [17], to counter SSL. Their research leads to several questions: (1) Are other DAAs designed for SL also ineffective against SSL (2) Does the proposed CP method maintain similar effectiveness in SL scenarios? (3) Can we develop a DAA capable of conducting effective attacks in both SL and SSL scenarios?

To answer these questions, we experimentally evaluate seven representative DAAs designed for SL on SSL. SSL is a strong defense against most DAAs due to its augmentation invariance objective; DAA features that are distorted or obfuscated by augmentations are ignored by the SSL algorithm and cannot affect downstream tasks. Accordingly, we find in Fig. 1 that most poisons for CIFAR-10 [23] fail to generalize to SSL. Only CUDA [27] and CP [17] exhibit significant impact on SSL. CUDA filters inject poison features throughout images that are resistant to cropping, flipping, and color shifts, and CP is designed specifically to counter SSL. We note a significant loss of performance for CP in SL scenarios. Therefore, CUDA emerges as the most potent DAA in both SSL and SL. However, the clean test accuracy of SSL trained on CUDA still hovers around 70%, suggesting that models trained on poisoned data retain some usability. As suggested by Sadasivan et al. [27], increasing the size of the convolution kernels in CUDA could provide a stronger poisoning effect, but it also degrades image quality. Fig. 2 shows that increased kernel size results in obviously blurred images. This compromise is often unacceptable in practical scenarios, particularly for sharing images and artwork online.

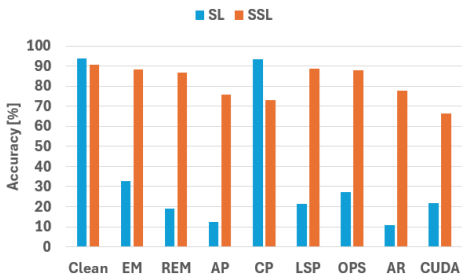


Fig. 1: The clean test accuracies of different DAAs on CIFAR-10 [23]. Lower accuracy represents better attack performance.

Fig. 2: The poisoned images generated by CUDA with different kernel sizes.

To comprehend CUDA's efficacy, Sadasivan et al. provide a theoretical analysis under assumptions of Gaussian-distributed data with independent elements and a two-class scenario. However, these assumptions may not hold in real-world scenarios. In addition, their analysis is not based on deep learning, making it challenging to derive insights for designing a more effective poison method.

To gain a deeper understanding of CUDA's impact on model training, we first conduct a theoretical analysis of CUDA using a deep learning model. Our analysis reveals how CUDA generates sub-optimal gradients for clean data and introduces class-wise bias through random filters. Building upon this analysis, we propose a new poisoning method, named Imperfect Restoration Poisoning (IRP), which maintains high image quality while achieving a stronger poisoning effect. We compare IRP to eight representative DAAs in both SL and SSL scenarios. In the SL scenario, we additionally compare IRP with other DAAs under various defense mechanisms, including adversarial training [33], Image Shortcut Squeezing [24], Mixup [44], Cutout [9], and CutMix [43]. All experimental results indicate that IRP significantly outperforms previous DAAs. Our main contributions are:

- We experimentally assess eight representative DAAs in both supervised and self-supervised learning scenarios, revealing that existing DAAs fail to achieve satisfactory effectiveness simultaneously in both contexts.
- We conduct a theoretical analysis of CUDA on deep learning models, revealing how it generates sub-optimal gradients for clean data and introduces class-wise bias through random filters.
- We propose a novel DAA named IRP based on imperfect restoration, which achieves high effectiveness in both SL and SSL and maintains superior image quality compared to CUDA.
- We perform an exhaustive comparison between IRP and eight other DAAs across SL, SSL, and five defense techniques on CIFAR-10 and a subset of ImageNet [8]. IRP outperforms baseline poisons by a large margin in all cases. We further verify IRP across five additional architectures and two supplementary datasets. IRP displays high effectiveness across all experiments.

2 Related Works

DAAs have garnered significant attention in recent years for their application to data protection. Various methodologies have been proposed to deliberately induce malfunctions in the target model in order to safeguard against unauthorized data usage. DAAs can be broadly categorized into two types, model-reliant methods [12–14, 17, 20, 42] and model-free methods [27, 29, 40, 41].

Model-Reliant Methods: Early approaches to DAAs often frame the poisoning problem as a bilevel optimization task, balancing loss minimization with respect to model parameters with loss maximization with respect to perturbed inputs. While these methods [1, 22] initially showed promise, their applicability to deep neural networks is limited due to intractability in obtaining exact solutions. Efforts to address optimization challenges in perturbation generation, as seen in works by Feng et al. [12] and Yuan et al. [42], often introduce significant computational overhead, limiting scalability for real-world applications. Alternatively, Huang et al. [20] propose another bi-level Error Minimizing (EM) approach, deliberately optimizing perturbations to minimize training loss. By alternately training a surrogate model and optimizing perturbations, the efficacy of the perturbations is attained through multiple rounds of optimization. In contrast to traditional

bi-level approaches, Fowl et al. [13] demonstrate the effectiveness of utilizing common objectives from adversarial examples for generating potent Adversarial Poisons (AP). However, Tao et al. [33] pinpoint that these DAAs can be easily broken by adversarial training. In response, Fu et al. [14] introduce Robust Error-Minimizing (REM) which enhances the poisoning effects against adversarial training by replacing the surrogate in EM with an adversarially-trained model. Recently, He et al. [17] assert that all aforementioned methods are tailored for supervised scenarios, thereby making them unsuitable for self-supervised learning. To address this gap, they introduce Contrastive Poison (CP), specifically designed for self-supervised learning. However, their assessment of DAAs tailored for SL is exclusively centered on AP, with other DAAs left unexplored. Additionally, while they show some effectiveness of their class-wise CP on both SL and SSL, the test accuracies on SSL remain high, ranging from 60.7% to 68.0%. In addition, class-wise additive perturbations can be recovered by the average image of a class, making them relatively easy to remove [29]. Therefore, we do not consider the class-wise poisons that apply the same additive noise to images, including class-wise attacks by EM, AP, and CP.

Model-Free Methods: Model-free DAAs tackle the data protection issue by leveraging shortcut learning. These methods are grounded in the observation that Deep Neural Networks (DNNs) often prioritize easily learnable shortcuts over semantic features, allowing them to distinguish between examples from different classes [3, 21]. Expanding on this concept, Yu et al. [41] propose incorporating artificially generated Linearly Separable Patterns (LSP) sampled from high-dimensional Gaussian distributions into images as an effective method for data poisoning. Sandoval-Segura et al. [29] discover that applying additive perturbations derived from autoregressive (AR) processes to clean data can serve as an effective shortcut. Wu et al. [40] examine the model's susceptibility to sparse poisons and demonstrate that consistently perturbing only one pixel is adequate to generate potent poisons. In contrast to applying additive perturbations, Sadasivan et al. [27] introduce a Convolution-based Unlearnable Dataset (CUDA) generation method, which utilizes randomly generated class-wise convolutional filters. In our evaluation, CUDA demonstrates moderate robustness to SSL. However, its effectiveness on SSL remains insufficient and CUDA grapples with a severe trade-off between image quality and poison efficacy.

DAA Mitigation Strategies: Previous studies have demonstrated that Adversarial Training (AT) [15] is an effective method to mitigate the potency of DAAs [33,39]. However, AT is impeded by its significant computational demands. Alternatively, various image preprocessing techniques, such as data augmentation (e.g., Mixup [44], Cutout [9], and CutMix [43]), have been explored. While these augmentation-based techniques show some impact, they are not as effective as AT [13]. Recently, Liu et al. [24] introduce Image Shortcut Squeezing (ISS) [24], a compression-based approach to counter DAAs. They demonstrate that ISS surpasses previously studied data augmentation countermeasures and achieves results comparable to AT. We evaluate the proposed IRP against multiple countermeasures, including AT, ISS, Mixup, Cutout, and CutMix.

3 Analysis

3.1 Threat Model

Like CUDA, most DAA papers define an "attacker" that applies a DAA on a labeled dataset that they want to protect yet publish. The attacker assumes no knowledge of any downstream DNN training process. It is therefore crucial to demonstrate that DAAs are effective across multiple training methods, including SL, SSL, and AT. The attacker generates class-based perturbations to their image set in order to inject poison features. The attacker publishes their poisoned images, optionally with captions or labels. Model trainers collect these poisoned images into datasets and may further alter or label the images.

3.2 Notation and Preliminaries

For clarity, we define a set of notations. f represents a Convolution Neural Network (CNN), including its training loss function but excluding the first layer convolution 2D filter $W \in \mathbb{R}^{d_1 \times d_2}$. $X \in \mathbb{R}^{m_1 \times m_2}$ denotes an input image and X_c represents an image belonging to class $c \in [1, ..., \mathbb{C}]$, where $\mathbb{C}$ is total number of the classes. Given that cross-correlation is predominantly used in place of convolutions in practice, the first layer operation is denoted as $X \star W$, where $\star$ signifies cross-correlation. Note that W is a part of the CNN and $f(X \star W)$ is the loss value of X. To retain readability, we focus on gradient analysis of the first layer filter, W. CUDA employs class-specific convolutional filters for data poisoning, and we provide a brief overview of its operation. Let $R_c \in \mathbb{R}^{\kappa \times \kappa}$ be a CUDA filter for class c images. According to [27], the generation procedure for R_c involves setting one random parameter among the κ^2 parameters to 1 within each R_c. Concurrently, the remaining parameters are initialized randomly, drawn from a uniform distribution with support $[0, p_u]$. An image X_c belonging to class c is poisoned by cross-correlation with the corresponding CUDA filter R_c, denoted as $X_c \star R_c$.

3.3 The Non-Optimal Gradient

To understand how CUDA lowers the accuracy of clean data, we analyze the gradient of f for both clean and CUDA-poisoned data. Without loss of generality, the training set consists of $\mathbb{C}$ clean images, each belonging to a different class, with corresponding poisoned images denoted as $X_i \star R_i$, $i \in [1, ..., \mathbb{C}]$. The gradient of the clean training data is

$$\frac{\partial \sum_{i=1}^{\mathbb{C}} f(X_i \star W)}{\partial W} = \sum_{i=1}^{\mathbb{C}} X_i \star \frac{\partial f(u_i)}{\partial u_i}, \tag{1}$$

where u_i is an intermediate variable equal to $X_i \star W$. The gradient of the poisoned training data is

$$\frac{\partial \sum_{i=1}^{\mathbb{C}} f((X_i \star R_i) \star W)}{\partial W} = \sum_{i=1}^{\mathbb{C}} (X_i \star R_i) \star \frac{\partial f(u_{p,i})}{\partial u_{p,i}}, \tag{2}$$

where the subscript p indicates that the intermediate variable is from poisoned data. Using these gradients to update W, we obtain

$$\begin{aligned} W_{t+1} &= W_t - \lambda \sum_{i=1}^{\mathbb{C}} X_i \star \frac{\partial f(u_i)}{\partial u_i} \\ W_{p,t+1} &= W_t - \lambda \sum_{i=1}^{\mathbb{C}} (X_i \star R_i) \star \frac{\partial f(u_{p,i})}{\partial u_{p,i}} \end{aligned} \tag{3}$$

where λ is learning rate. Inputting all clean training images, $X_1, \cdots, X_C$ to $f(\cdot \star W_{t+1})$ and $f(\cdot \star W_{p,t+1})$, we obtain clean and poison loss values $\sum_{i=1}^{\mathbb{C}} f\left(X_i \star W_{t+1}\right)$ and $\sum_{i=1}^{\mathbb{C}} f\left(X_i \star W_{p,t+1}\right)$, respectively. Given that $\sum_{i=1}^{\mathbb{C}} X_i \star \frac{\partial f(o_i)}{\partial i}$ in Eq. 1 is the optimal gradient descent direction for the clean data, any deviation from this direction will result in increased loss when λ is small. Thus, $\sum_{i=1}^{\mathbb{C}} f\left(X_i \star W_{p,t+1}\right) > \sum_{i=1}^{\mathbb{C}} f\left(X_i \star W_{t+1}\right)$ for clean data. In other words, CUDA employs a non-optimal gradient to disrupt training, rendering clean data unrecognizable.

3.4 Poisoning through Class-Wise Bias

Though the previous subsection uncovers how CUDA filters increase training loss, it does not indicate how these filters poison the network. To further investigate its mechanism, we input $X_c \star R_c$ to $f(\cdot \star W_{p,t+1})$, whose first layer output is

$$(X_c \star R_c) \star W_{p,t+1} = (X_c \star R_c) \star W_t - \lambda (X_c \star R_c) \star \sum_{i=1}^{C} \left((X_i \star R_i) \star \frac{\partial f(u_{p,i})}{\partial u_{p,i}} \right). \tag{4}$$

Assuming W_t is not poisoned by CUDA filters, we ignore the term $(X_c \star R_c) \star W_t$. Since cross-correlation is equivalent to a rotated convolution (i.e., $X_c \star R_c = X_c * R_{c\pi}$, where $*$ represents convolution and $R_{c\pi}$ represents rotating the 2D filter R_c by π), the second term of Eq. 4 can be rewritten as

$$\lambda X_c * \sum_{i=1}^{C} R_{c\pi} * R_i * X_{i\pi} * \frac{\partial f(u_{p,i})}{\partial u_{p,i}}. \tag{5}$$

Eq. 5 shows that the output of the first layer is influenced by $R_{c\pi} * R_i$. Since $R_{c\pi} * R_i = R_{c\pi} \star R_{i\pi}$, we analyze $R_{c\pi} \star R_{i\pi}$. Theorem 1 describes the statistical probabilities of $R_{c\pi} \star R_{i\pi}$ and $R_{c\pi} \star R_{c\pi}$. The proof is provided in Appendix A.1.

Theorem 1. *Given two different filters, R_i and R_c, generated by CUDA, if $1 > (\kappa^2 - 2)p_u^2 + 2p_u$, then $R_{c\pi} \star R_{i\pi}$ and $R_{c\pi} \star R_{c\pi}$ have the following properties:*

1. *The peak of $R_{c\pi} \star R_{c\pi}$ is located at the center.*
2. *The peak of $R_{c\pi} \star R_{i\pi}$ occurs at the position where two ones in $R_{c\pi}$ and $R_{i\pi}$ appear at the same position in $R_{c\pi} \star R_{i\pi}$.*
3. *The probability that the peak of $R_{c\pi} \star R_{i\pi}$ lies in the center is $\frac{1}{\kappa^2}$.*
4. *The expected peak of $R_{c\pi} \star R_{c\pi}$ is higher than the expected peak of $R_{c\pi} \star R_{i\pi}$.*

Figure 3 presents two CUDA filters along with the corresponding $R_{c\pi} \star R_{c\pi}$ and $R_{c\pi} \star R_{i\pi}$ to illustrate these statistical properties. In practice, CUDA utilizes $p_u = 0.06$ and $\kappa = 9$ for ImageNet, which fulfills $1 > (\kappa^2 - 2)p_u^2 + 2p_u$. Similar properties also apply on CIFAR-10 (see Appendix A.3). These properties indicate that $R_{c\pi} * R_c$ in Eq. 5 retains more information in $X_{c\pi} * \frac{\partial f(u_{p,c})}{\partial u_{p,c}}$ but $R_{c\pi} * R_i$ shifts the information in $X_{i\pi} * \frac{\partial f(u_{p,i})}{\partial u_{p,i}}$ to a different position with a probability of $(\kappa^2 - 1)/\kappa^2$. For $\kappa = 9$, the probability is 0.988. Consequently, the upper layers receive class c information in the correct position, but information from other classes is shifted to incorrect positions. For $p_u = 0.06$, we note the peak value of $R_{c\pi} \star R_{i\pi}$ is smaller than the sum of its non-peak values (estimated mean ratio of 0.099). This means that non-peak values also play an important role in degrading other class information. This blurring effect also affect the c-class, though to a lesser extent. More clearly, the estimated mean ratio of the peak value of $R_{c\pi} \star R_{c\pi}$ over the sum of its non-peak values is 0.106. Combining these two effects, CUDA filters utilize peak shifts and unique blur patterns to introduce class-wise bias in the poisoned class.

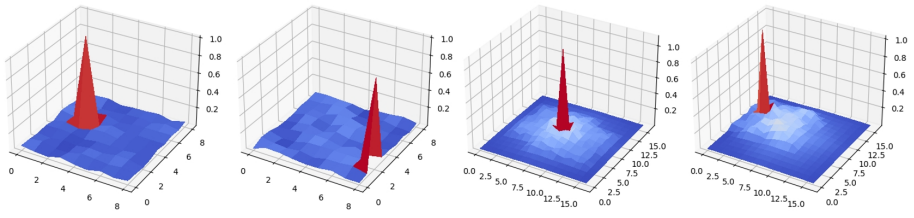


Fig. 3: Two CUDA filters $R_{c\pi}$ and $R_{i\pi}$(the first and second plots) along with the corresponding $R_{c\pi} \star R_{c\pi}$ (the third plot) and $R_{c\pi} \star R_{i\pi}$ (the last plot).

3.5 Poisoning with Imperfect Restoration

The previous analysis shows that CUDA filters create class-wise bias, and that $R_{c\pi} \star R_{c\pi}$ retains the training information of class c. When $X_c \star R_c$ is closer to X_c, it is expected to preserve the class c information better and enhance the image quality. However, if $X_c \star R_c = X_c$, then there is no poisoning effect. Thus, we propose the Imperfect Restoration Poison (IRP). Let X_{ci} be one of the class c images, where $i \in 1, \cdots, n$. IRP first applies CUDA filter R_c on all class c training images and obtains $X_{ci} \star R_c$. Then, the filtered images are cropped into patches with the size same as the CUDA filter i.e., $\kappa \times \kappa$, denoted as $(X_{ci} \star R_c)_j$. The pixel value of X_{ci} corresponding to the center of the patch is denoted y_{cij}, and the patch is reshaped into a column vector denoted as η_{cij}. Finally, a class-wise linear model $\alpha_c \in \mathbb{R}^{\kappa^2 \times 1}$ and the mean square loss are used to estimate the value of y_{cij}. More precisely, the class-wise model $\widehat{\alpha_c}$ is obtained by minimizing

$$\widehat{\alpha_c} = \underset{\alpha_c}{\operatorname{argmin}} \sum_i \sum_j \|y_{cij} - \alpha_c^T \eta_{cij}\|_2^2 \tag{6}$$

Once $\widehat{\alpha_c}$ is obtained, it is reshaped back to a $\kappa \times \kappa$ filter, denoted as A_c. Finally, IRP uses $X_{ci} \star P_c$ to perform poisoning, where $P_c = R_c \star A_c$. Theorem 2 establishes that IRP is incapable of achieving perfect restoration (proof in Appendix A.2).

Theorem 2. *Given a CUDA filter size larger than 1 and x_{cij} as the vector form of the patch in X_{ci} involved in producing η_{cij} (i.e., $\eta_{cij} = \wp x_{cij}$, where $\wp$ is a $\kappa^2 \times (3\kappa-2)^2$ matrix), if the column vectors of $\wp$, excluding the one corresponding to y_{cij}, span $\mathbb{R}^{\kappa^2}$, then, $\exists x_{cij} \in \mathbb{R}^{(3\kappa-2)^2}$ such that $y_{cij} \neq a_c^T \wp x_{cij}$.*

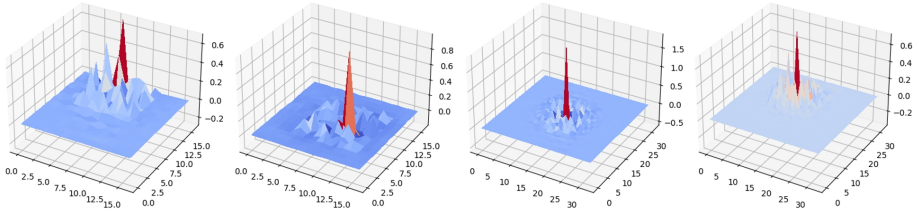
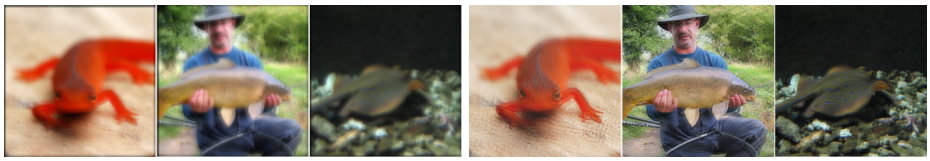


Fig. 4: Two IRP filters $P_{c\pi}$ and $P_{i\pi}$ (the first and second plots) along with the corresponding $P_{c\pi} \star P_{c\pi}$ (the third plot) and $P_{c\pi} \star P_{i\pi}$ (the last plot).

Fig. 4 shows two IRP filters with the corresponding self-correlation and cross-correlation. Similar to CUDA, we observe that $P_{c\pi} \star P_{c\pi}$ exhibits a peak at the center, while $P_{c\pi} \star P_{i\pi}$ has a peak in a different position. However, they have more complex patterns around the peaks. Therefore, like CUDA, IRP exploits peak shifts to create class-wise bias. However, IRP filters have turbulent patterns around the peaks, creating unique high-frequency attack patterns that are more complex than the low-frequency patterns from CUDA. Due to their dependence on the training data, it would be challenging to theoretically analyze IRP filters, as was done for CUDA, without proper assumptions on the data distribution. We refrain from imposing unrealistic assumptions on the data distribution (e.g., Gaussian) as this could lead to misleading analytic results.

Fig. 5 displays some images generated by IRP and CUDA. IRP images are visually clearer and maintain higher quality. Notably, the man's face is sharper and the spots of the salamander and stingray are easily distinguishable under IRP protection. The backgrounds of all three images are sharper under IRP.



(a) CUDA **(b)** IRP

Fig. 5: Poisoned examples generated by CUDA (left) and IRP (right).

4 Experiments

In this section, we begin by evaluating the effectiveness of IRP alongside eight DAA methods in standard SL and SSL settings, as well as against representative defense methods. We then investigate the effectiveness of different DAAs under partial poisoning scenarios. For model-reliant methods, our experiments primarily focus on the nominal setup, wherein a surrogate model generates poison images. To assess the transferability of IRP, we evaluate IRP across six different architectures on three datasets. Finally, we compare the image quality generated by CUDA and IRP and ablate different filter types across various training methods.

4.1 Experimental Setup

Datasets: We examine poisoning on CIFAR-10 [23], CIFAR-100 [23], STL-10 [7], and ImageNet-100, a subset of ImageNet [8] consisting of 100 classes. By default, we use 50,000 images for training and 10,000 images for testing on CIFAR-10 and CIFAR-100. For STL-10, we use their default 5000 training images and 8000 test images for training and testing, respectively. For ImageNet-100, we train on the images from the first 100 classes of the official training set and test on all corresponding images from the official validation set.

Poison Settings: We compare IRP with eight state-of-the-art DAAs, including four model-reliant methods, including EM [20], TAP [13], REM [14], and CP [17], and four model-free methods, including CUDA [27], LSP [41], OPS [40], and AR [29]. As generating poison data using model-reliant methods is time-consuming on ImageNet, we generate a 20% subset of ImageNet-100 for CP, TAP, and EM. We maintain all default settings and retain the L_∞ norm of the defensive perturbation at $8/255$. We utilize the full REM-poisoned ImageNet-100 dataset shared by the REM authors. According to [14], the L_∞ norm of the defensive perturbation ρ_u is set to $8/255$, and the adversarial perturbation radius ρ_a is set to $4/255$ for controlling the level of protection of the noise against adversarial training. For model-free methods, we maintain the settings as per their original papers, except for LSP and OPS on ImageNet-100. As the original settings could not yield a satisfactory poisoning effect, we amplify the perturbation of LSP and OPS to $L_\infty = 16/255$ and $L_0 = 5$, respectively. As AR does not report its effectiveness on ImageNet-100 and has shown to be vulnerable to SSL and AT on CIFAR-10, we opt not to include AR in further testing on ImageNet-100.

Models: Unless stated otherwise, we employ ResNet-18 [18] as the surrogate for model-reliant methods and as the target models for all the testing methods in both SL and SSL scenarios. Various SSL algorithms are employed, including SimCLR [4], SimSiam [5], MoCoV3 [6] and BYOL [16]. Linear probing accuracy, consistent with CP [17], is utilized to evaluate the effectiveness of DAAs in SSL. To assess transferability, we evaluate IRP on target models with various architectures, including ResNet-18, ResNet-34 [18], VGG-19 [32], DenseNet-121 [19], MobileNet-V2 [28], and ViT [11]. Training details are deferred to Appendix B.1.

Table 1: Top-1 clean test accuracies (%) of ResNet-18 trained with data poisoned by various DAAs on CIFAR-10. The "Max" column represents the highest clean test accuracy, which corresponds to the worst performance of each DAA under different training scenarios. Bold indicates the best performance, and underline denotes the second-best performance.

DAA	SL	SSL	AT	Cutout	Cutmix	Mixup	ISS	Max
Clean	93.86	90.55	89.57	95.62	95.78	95.46	82.40	95.62
EM	32.53	88.45	89.31	36.18	38.40	54.19	82.12	89.31
REM	18.91	86.70	34.45	19.67	27.46	23.27	80.04	86.70
AP	12.45	75.77	85.67	**9.70**	**8.38**	33.27	81.10	85.67
CP	93.59	73.08	88.43	94.51	93.77	93.36	82.21	94.51
LSP	21.47	88.87	85.67	18.47	21.68	22.76	79.33	88.87
OPS	27.24	87.79	**20.10**	65.12	85.27	37.56	72.14	87.79
AR	<u>10.95</u>	86.86	72.94	12.17	12.95	**14.15**	82.83	86.86
CUDA	21.89	<u>66.58</u>	48.58	23.46	24.04	21.75	**21.94**	<u>66.58</u>
IRP	**10.39**	**43.24**	<u>32.21</u>	<u>10.85</u>	<u>15.21</u>	<u>16.03</u>	<u>29.42</u>	**43.24**

4.2 Comparisons to Baseline DAAs

Effectiveness on SL and SSL: We first evaluate IRP and the other DAAs on standard SL and SSL training scenarios on CIFAR-10 and ImageNet-100. SimCLR is utilized for model pretraining in the SSL evaluation. Columns 2-3 of Tables 1 and 2 show that IRP achieves state-of-the-art poisoning, enforcing the lowest clean test accuracies in both SL and SSL and for both CIFAR-10 and ImageNet-100.

We begin by analyzing results for CIFAR-10 in Table 1. For SL with CIFAR-10, AR and IRP demonstrate a similar level of poisoning effect under SL, reducing the test accuracy to around 10%. However, AR underperforms on SSL, permitting a significantly higher accuracy of 86.86% compared to that of IRP at 43.24%. CUDA is the second-best DAA for SSL with an accuracy of 66.58%, which is still drastically higher than that of IRP.

For ImageNet-100 results in Table 2, we begin by nothing the drop in effectiveness of OPS relative to CIFAR-10. This is attributed to the RandomResizeCrop augmentation utilized for training models on ImageNet-100. We chose to use this augmentation on ImageNet-100 as it is commonly used for SL and SSL on ImageNet-100 in literature and it enhances the effectiveness of all the defense methods. For SL with ImageNet-100, CUDA, EM, AP, and LSP demonstrate strong poisoning capabilities, reducing the clean test accuracy to below 10%. Even so, IRP outperforms all others with a test accuracy of 1.98%. For SSL with ImageNet-100, the test accuracies of most DAAs remain high, except for CUDA at 21.89% and CP at 18.76%. IRP outperforms both CUDA and CP, achieving a clean test accuracy of 9.30%. These results establish IRP as the most effective DAA under both SL and SSL scenarios.

Table 2: Top-1 clean test accuracies (%) of ResNet-18 trained with data poisoned by various DAAs on ImageNet-100. The "Max" column represents the highest clean test accuracy, which corresponds to the worst performance of each DAA under different training scenarios. Bold indicates the best performance, and underline denotes the second-best performance.

DAA	SL	SSL	AT	Cutout	Cutmix	Mixup	ISS	Max
Clean	77.66	70.96	69.76	78.04	81.08	80.38	71.58	80.38
EM	6.32	50.02	43.26	6.42	5.80	12.38	41.18	50.02
REM	14.18	65.30	57.34	15.60	16.10	33.08	67.52	67.52
AP	8.18	41.80	42.88	7.68	8.38	9.68	24.22	42.88
CP	57.46	<u>18.76</u>	48.76	58.20	62.28	61.76	50.18	62.28
LSP	6.62	62.06	<u>28.92</u>	<u>5.06</u>	7.84	5.50	33.70	62.06
OPS	51.00	62.22	52.56	53.86	65.12	48.82	57.36	65.12
CUDA	<u>6.10</u>	26.12	36.34	8.66	<u>5.70</u>	<u>5.16</u>	**3.58**	<u>36.34</u>
IRP	**1.98**	**9.30**	**22.10**	**2.18**	**1.52**	**2.02**	<u>4.14</u>	**22.10**

Effectiveness under Common Countermeasures: To further evaluate the effectiveness of IRP under defensive scenarios, we test IRP and the baselines on representative defense methods, including Adversarial Training (AT) [15], Image Shortcut Squeezing (ISS) [24], and common data augmentation techniques such as Cutout, CutMix, and Mixup. Following [14], we employ the L_∞ norm for AT and set the norm bound to $4/255$. 10-step PGD with a step size of $0.6/255$ is used for AT. As suggested by [24], we employ a combination of grayscale and JPEG with the default JPEG quality factor (JPEG-10) for ISS to ensure global effectiveness against all DAAs. The results on CIFAR-10 and ImageNet-100 are presented in columns 4-8 of Tables 1 and 2, respectively. The last column displays the maximum clean test accuracy of the corresponding methods across all testing scenarios, representing the worst performance among all the test cases.

DAA results on CIFAR-10 in Table 1 vary across defenses. For instance, OPS shows high effectiveness on AT but is susceptible to ISS and CutMix, while AP performs well against CutMix and Cutout but is less effective against AT and ISS. Only CUDA and IRP demonstrate effectiveness across all countermeasures. However, CUDA permits maximum clean accuracies of 48.58% across all countermeasures and 66.58% across all testing cases, which are 16.37% and 23.24% higher, respectively, than the 43.24% maximum accuracy allowed by IRP.

Similarly, CUDA and IRP emerge as the two most effective methods across all defenses on ImageNet-100 in Table 2. However, CUDA permits a maximum clean accuracy of 36.34%, significantly higher than that of IRP at 22.10%. The results demonstrate that IRP is the most effective DAA in all testing scenarios.

Evaluation on Other SSL Algorithms: To further assess the efficacy of IRP on SSL, we test IRP-poisoned ImageNet-100 on three additional SSL algorithms: SimSiam [5], MoCoV3 [6], and BYOL [16]. As shown in Table 3, IRP lowers the test accuracies of all SSL algorithms to around 10%. Even in its the worst case performance with BYOL, IRP reduces clean accuracy to 11.6%.

Table 3: Top-1 accuracies (%) of ResNet-18 trained with different SSL algorithms on ImageNet-100.

	SimCLR	MoCoV3	SimSiam	BYOL	Max
Clean	70.96	75.54	70.68	75.34	75.54
IRP	9.3	10.58	7.82	11.6	11.6

4.3 Additional Training Scenarios and Datasets

Different Poison Percentages: In this section, we first evaluate IRP on a more challenging and realistic learning scenario, where only a portion of the data is shielded by IRP. We randomly select a portion of the training data from the entire training dataset of CIFAR-10 and apply IRP to the selected subset. The poisoned data is denoted as D_p and the clean data in the training dataset that has not been used for generating poison data is denoted as D_c. Then, we conduct standard supervised training with ResNet-18 on the mixed data (D_{p+c}) consisting of the D_p and D_c, as well as on the clean data subset D_c. The results in Table 4 show that the effectiveness drops quickly when the data are not 100% poisoned. This phenomenon applies for all other DAAs as well. Models trained only on D_c demonstrate a similar level of performance as the models trained with D_{p+c}. This suggests that the high performance for <100% poisoning is not due to a failure of the DAAs. Finally, works on DAAs for artwork protection [30, 31] have noted similar trends but find that individuals can still protect their own data via DAA even if the majority of artworks are unprotected. That is, it's still possible to make subsets of the training data unlearnable.

Table 4: The top-1 clean test accuracies (%) of ResNet-18 trained with partial poisoned data on CIFAR-10. The percentage in the first row indicates the percentage of poison data, and D_c indicates the remaining cleaning data. 0% and 100% signify training scenarios where the entire dataset remains either clean or fully poisoned, respectively.

	0%	20%		40%		60%		80%		100%
		D_{p+c}	D_c	D_{p+c}	D_c	D_{p+c}	D_c	D_{p+c}	D_c	
EM	93.86	94.02	92.38	92.72	91.74	90.71	87.84	85.75	77.36	32.53
REM		92.85		92.15		90.05		86.29		18.91
AP		92.57		**91.16**		90.72		88.11		12.45
LSP		92.95		92.40		90.19		85.76		21.47
OPS		93.40		92.25		90.63		86.68		27.24
AR		**92.98**		92.66		90.94		87.93		10.95
CUDA		93.27		92.27		90.51		86.90		21.89
IRP		93.07		92.56		**89.89**		**85.70**		**10.39**

Effectiveness on Different Models: The previous evaluations are all conducted on ResNet-18 (RN18). To evaluate the effectiveness of IRP on different

Table 5: The top-1 clean test accuracies (%) of different models trained on data poisoned by IRP and CUDA with AT.

Dataset		RN18	RN34	VGG19	DN121	MNv2	ViT-S	Max
CIFAR10	Clean	89.57	89.32	78.57	79.97	75.94	47.84	89.57
	CUDA	48.58	43.63	42.02	49.44	26.33	44.93	49.44
	IRP	**32.21**	**17.37**	**19.06**	**17.87**	**14.13**	**35.39**	**35.39**
CIFAR100	Clean	65.30	66.43	48.50	58.92	47.00	27.19	66.43
	CUDA	37.21	37.95	27.44	**23.07**	**16.00**	24.88	37.95
	IRP	**27.13**	**28.85**	**22.50**	25.24	20.70	**20.82**	**28.85**
STL10	Clean	71.94	69.73	64.35	72.41	58.93	43.86	72.41
	CUDA	44.70	44.35	45.99	39.39	44.49	42.14	45.99
	IRP	**34.10**	**34.33**	**33.89**	**33.19**	**33.81**	**35.58**	**35.58**

architectures, we evaluate ResNet-34 (RN34), VGG19, DenseNet121 (DN121), MobileNetv2 (MNv2), and ViT-S on IRP-poisoned CIFAR-10. All models are trained with AT (L_∞ constraint of 4/255, as before), which emerged as the strongest defense against different DAAs from our experiments in Section 4.2. We include CUDA in this evaluation as a reference. On all the examined networks, IRP outperforms CUDA by a significant margin. In their respective worst cases, CUDA permits a test accuracy of 49.44%, while IRP allows a test accuracy of 35.39%. Table 5 shows that even under AT defense, IRP is still effective on all the examined architectures.

Evaluation on Different Datasets: We extend our evaluation to the CIFAR-100 and STL-10 datasets. The same set of models and AT settings are utilized. As shown in Table 5, both IRP and CUDA demonstrate effectiveness on CIFAR-100 and STL-10 across all the network architectures under AT. IRP consistently outperforms CUDA by a large margin on most networks for both CIFAR-100 and STL-10. In the two cases where IRP is weaker than CUDA, the clean test accuracies of IRP are still within 5% of those of CUDA. The maximum clean test accuracies of IRP on CIFAR-100 and STL-10 are 28.85% and 35.58%, respectively. Both of these values are significantly lower than those of CUDA, which are 37.95% and 45.99% on CIFAR-100 and STL-10, respectively. This experiment demonstrates that IRP is universally effective across networks and across different datasets.

4.4 Image Quality Comparison against CUDA

We compare the quality of ImageNet-100 images poisoned by IRP and CUDA. We quantify image quality with three common full-reference quality indices, including LPIPS [45], SSIM [37], and MS-SSIM [38], and two no-reference quality indices, including CLIP-IQA [35] and BRISQUE [25]. The original image is used as the reference for LPIPS, SSIM, and MS-SSIM. Table 6 presents the evaluation results of all quality indices. Both Table 6 and Fig. 5 demonstrate that IRP generates poisoned images with better image quality than CUDA.

Metric	Clean	CUDA	IRP
LPIPS ↓	-	0.272	0.142
SSIM ↑	-	0.561	0.775
MS-SSIM↑	-	0.831	0.903
CLIP-IQA↑	0.787	0.528	0.706
BRISQUE↓	16.625	29.570	17.318

Table 6: ImageNet-100 image quality under different quality indices.

Fig. 6: The Top-1 accuracy of different setting on ImageNet-100.

4.5 Filter Ablation

The IRP filter $P_c = R_c \star A_c$ is a kernel with a larger size of $(2\kappa + 1) \times (2\kappa + 1)$, where R_c and A_c are both filters of size $\kappa \times \kappa$. To verify that the high effectiveness of IRP is not caused by the bigger kernel size, we generate filters of the same size as the IRP under two different filter settings. For 'Random Sharpness', we generate a random sharpness filter A_{c1} of size $\kappa \times \kappa$ for each class. Within each A_{c1}, one random parameter out of the $\kappa \times \kappa$ parameters is set to 1, while the remaining parameters are randomly initialized from a uniform distribution with support $[-p_s, 0]$. After testing several values for p_s, we find that 0.01 is suitable for generating poisoned images with good visual quality. We then poison class c images by $X_c \star R_c \star A_{c1}$. For 'Double Blur', we generate another random CUDA filter R_{c1} of size $\kappa \times \kappa$ for each class, and then poison class c images by $X_c \star R_c \star R_{c1}$.

We compare the effectiveness of the two additional filter settings with CUDA and IRP on SL, SSL, and AT for ImageNet-100. The results in Fig. 6 show that Random Sharpness and Double Blur exhibit slightly better performance than CUDA in SSL. However, the effectiveness of Random Sharpness and Double Blur cannot match that of IRP. Additionally, we provide examples of images generated by Double Blur and Random Sharpness in Appendix C.1. These samples highlight their inferior quality compared to those generated by IRP.

5 Conclusion

In this study, we conduct a theoretical analysis of CUDA, uncovering the suboptimal gradients it introduces and elucidating the strategy it employs to induce class-wise bias for data poisoning. Building on these insights, we introduce IRP, which demonstrates high effectiveness in both SL and SSL as well as under state-of-the-art defense scenarios. IRP also maintains high image quality, rendering it suitable for data protection in real-world applications.

Acknowledgements

This research is supported by the National Research Foundation, Singapore and Infocomm Media Development Authority under its Trust Tech Funding Initiative and Strategic Capability Research Centres Funding Initiative. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore and Infocomm Media Development Authority.

Mingzhi Lyu and Fan Wang are supported by ROSE @ NTU, Interdisciplinary Graduate Programme, Nanyang Technological University, Singapore.

References

1. Biggio, B., Nelson, B., Laskov, P.: Poisoning attacks against support vector machines. arXiv preprint arXiv:1206.6389 (2012)
2. Birhane, A., Prabhu, V.U.: Large image datasets: A pyrrhic win for computer vision? In: 2021 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 1536–1546. IEEE (2021)
3. Caron, E., Chrétien, S.: A finite sample analysis of the benign overfitting phenomenon for ridge function estimation. arXiv preprint arXiv:2007.12882 (2020)
4. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. arXiv preprint arXiv:2002.05709 (2020)
5. Chen, X., He, K.: Exploring simple siamese representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 15750–15758 (2021)
6. Chen*, X., Xie*, S., He, K.: An empirical study of training self-supervised vision transformers. arXiv preprint arXiv:2104.02057 (2021)
7. Coates, A., Ng, A., Lee, H.: An analysis of single-layer networks in unsupervised feature learning. In: Proceedings of the fourteenth international conference on artificial intelligence and statistics. pp. 215–223. JMLR Workshop and Conference Proceedings (2011)
8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
9. DeVries, T., Taylor, G.W.: Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552 (2017)
10. Dolatabadi, H.M., Erfani, S., Leckie, C.: The devil's advocate: Shattering the illusion of unexploitable data using diffusion models (2024), `https://arxiv.org/abs/2303.08500`
11. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
12. Feng, J., Cai, Q.Z., Zhou, Z.H.: Learning to confuse: generating training time adversarial data with auto-encoder. Advances in Neural Information Processing Systems **32** (2019)
13. Fowl, L., Goldblum, M., Chiang, P.y., Geiping, J., Czaja, W., Goldstein, T.: Adversarial examples make strong poisons. Advances in Neural Information Processing Systems **34**, 30339–30351 (2021)

14. Fu, S., He, F., Liu, Y., Shen, L., Tao, D.: Robust unlearnable examples: Protecting data privacy against adversarial learning. In: International Conference on Learning Representations (2022), https://openreview.net/forum?id=baUQQPwQiAg
15. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
16. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al.: Bootstrap your own latent-a new approach to self-supervised learning. Advances in neural information processing systems **33**, 21271–21284 (2020)
17. He, H., Zha, K., Katabi, D.: Indiscriminate poisoning attacks on unsupervised contrastive learning. In: The Eleventh International Conference on Learning Representations (2023), https://openreview.net/forum?id=f0a_dWEYg-Td
18. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
19. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4700–4708 (2017)
20. Huang, H., Ma, X., Erfani, S.M., Bailey, J., Wang, Y.: Unlearnable examples: Making personal data unexploitable. arXiv preprint arXiv:2101.04898 (2021)
21. Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., Madry, A.: Adversarial examples are not bugs, they are features. Advances in neural information processing systems **32** (2019)
22. Jagielski, M., Oprea, A., Biggio, B., Liu, C., Nita-Rotaru, C., Li, B.: Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In: 2018 IEEE symposium on security and privacy (SP). pp. 19–35. IEEE (2018)
23. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
24. Liu, Z., Zhao, Z., Larson, M.: Image shortcut squeezing: Countering perturbative availability poisons with compression. arXiv preprint arXiv:2301.13838 (2023)
25. Mittal, A., Moorthy, A.K., Bovik, A.C.: No-reference image quality assessment in the spatial domain. IEEE Transactions on image processing **21**(12), 4695–4708 (2012)
26. Qin, T., Gao, X., Zhao, J., Ye, K., Xu, C.Z.: Learning the unlearnable: Adversarial augmentations suppress unlearnable example attacks (2023), https://arxiv.org/abs/2303.15127
27. Sadasivan, V.S., Soltanolkotabi, M., Feizi, S.: Cuda: Convolution-based unlearnable datasets. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3862–3871 (2023)
28. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4510–4520 (2018)
29. Sandoval-Segura, P., Singla, V., Geiping, J., Goldblum, M., Goldstein, T., Jacobs, D.: Autoregressive perturbations for data poisoning. Advances in Neural Information Processing Systems **35**, 27374–27386 (2022)
30. Shan, S., Cryan, J., Wenger, E., Zheng, H., Hanocka, R., Zhao, B.Y.: Glaze: Protecting artists from style mimicry by text-to-image models (2023), https://arxiv.org/abs/2302.04222
31. Shan, S., Ding, W., Passananti, J., Wu, S., Zheng, H., Zhao, B.Y.: Nightshade: Prompt-specific poisoning attacks on text-to-image generative models (2024), https://arxiv.org/abs/2310.13828

32. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
33. Tao, L., Feng, L., Yi, J., Huang, S.J., Chen, S.: Better safe than sorry: Preventing delusive adversaries with adversarial training. Advances in Neural Information Processing Systems **34**, 16209–16225 (2021)
34. Vincent, J.: Getty images sues ai art generator stable diffusion in the us for copyright infringement (2023), https://www.theverge.com/2023/2/6/23587393/ai-art-copyright-lawsuit-getty-images-stable-diffusion
35. Wang, J., Chan, K.C., Loy, C.C.: Exploring clip for assessing the look and feel of images. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 2555–2563 (2023)
36. Wang, X., Hu, S., Li, M., Yu, Z., Zhou, Z., Zhang, L.Y.: Corrupting convolution-based unlearnable datasets with pixel-based image transformations (2024), https://arxiv.org/abs/2311.18403
37. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004)
38. Wang, Z., Simoncelli, E.P., Bovik, A.C.: Multiscale structural similarity for image quality assessment. In: The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003. vol. 2, pp. 1398–1402. Ieee (2003)
39. Wen, R., Zhao, Z., Liu, Z., Backes, M., Wang, T., Zhang, Y.: Is adversarial training really a silver bullet for mitigating data poisoning? In: The Eleventh International Conference on Learning Representations (2023), https://openreview.net/forum?id=zKvm1ETDOq
40. Wu, S., Chen, S., Xie, C., Huang, X.: One-pixel shortcut: On the learning preference of deep neural networks. In: The Eleventh International Conference on Learning Representations (2023)
41. Yu, D., Zhang, H., Chen, W., Yin, J., Liu, T.Y.: Availability attacks create shortcuts. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 2367–2376 (2022)
42. Yuan, C.H., Wu, S.H.: Neural tangent generalization attacks. In: International Conference on Machine Learning. pp. 12230–12240. PMLR (2021)
43. Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y.: Cutmix: Regularization strategy to train strong classifiers with localizable features. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 6023–6032 (2019)
44. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. In: International Conference on Learning Representations (2018)
45. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018)