Detecting and Corrupting Convolution-based Unlearnable Examples

Minghui Li[1*], Xianlong Wang[2,3*†], Zhifei Yu[3], Shengshan Hu[2,3], Ziqi Zhou[4], Longling Zhang[2,3], Leo Yu Zhang[5]

[1] School of Software Engineering, Huazhong University of Science and Technology
[2] Hubei Key Laboratory of Distributed System Security, Hubei Engineering Research Center on Big Data Security
[3] School of Cyber Science and Engineering, Huazhong University of Science and Technology
[4] School of Computer Science and Technology, Huazhong University of Science and Technology
[5] School of Information and Communication Technology, Griffith University
{minghuili,wxl99,yzf,hushengshan,zhouziqi,longlingzhang}@hust.edu.cn, leo.zhang@griffith.edu.au

Abstract

Convolution-based *unlearnable examples* (UEs) employ class-wise multiplicative convolutional noise to training samples, severely compromising model performance. This firenew type of UEs have successfully countered all defense mechanisms against UEs. The failure of such defenses can be attributed to the absence of norm constraints on convolutional noise, leading to severe blurring of image features. To address this, we first design an **E**dge **P**ixel-based **D**etector (**EPD**) to identify convolution-based UEs. Upon detection of them, we propose the first defense scheme against convolution-based UEs, **CO**rrupting these samples via random matrix multiplication by employing bilinear **IN**terpolation (**COIN**) such that disrupting the distribution of class-wise multiplicative noise. To evaluate the generalization of our proposed COIN, we newly design two convolution-based UEs called VUDA and HUDA to expand the scope of convolution-based UEs. Extensive experiments demonstrate the effectiveness of detection scheme EPD and that our defense COIN outperforms 11 *state-of-the-art* (SOTA) defenses, achieving a significant improvement on the CIFAR and ImageNet datasets.

Code — https://github.com/wxldragon/COIN

Introduction

The triumph of *deep neural networks* (DNNs) hinges on copious high-quality training data, motivating many commercial enterprises to scrape images from unidentified sources. In this scenario, adversaries may introduce elaborate and imperceptible perturbations to image data to serve as *unlearnable examples* (UEs) that are subsequently disseminated online. This leads to a diminished generalization capacity of the victim model after being trained on such samples (Huang et al. 2021; Meng et al. 2024). Previous UEs apply additive perturbations under the $\mathcal{L}_p$ norm constraint to ensure sample's visual concealment, referred to as *bounded UEs* (Huang et al. 2021; Fu et al. 2022; Liu, Wang, and Gao 2024). Correspondingly, many defense schemes (Tao et al. 2021; Wang et al. 2024a) have been proposed, largely compromising bounded UEs. The ease with which bounded UEs are successfully defended can be attributed to the fact that the introduced additive noise is limited, rendering the noise distribution easily disrupted. However, the latest proposed UE that employs convolution operations without norm constraints, known as *convolution-based UE* (Sadasivan, Soltanolkotabi, and Feizi 2023), expands the scope of noise as demonstrated in Fig. 1 (b), compromising the performance of current defenses, as demonstrated in Tab. 1.

Defense	Defending CUDA	Defending VUDA	Defending HUDA	Model-agnostic
Cutout	❍	❍	❍	●
Mixup	❍	❍	❍	●
Cutmix	❍	❍	❍	●
DP-SGD	❍	❍	❍	❍
AT	❍	❍	❍	❍
AA	❍	❍	❍	●
AVATAR	❍	❍	❍	❍
OP	❍	❍	❍	❍
ISS-G	❍	❍	❍	●
ISS-J	❍	❍	❍	●
ECLIPSE	❍	❍	❍	❍
COIN (Ours)	●	●	●	●

Table 1: The effectiveness of existing defenses against three convolution-based UEs and the dependence on models are presented. "●" denotes fully satisfying the condition.

To the best of our knowledge, none of the existing defense mechanisms can effectively defend against convolution-based UEs.

To address this, the first step is to detect convolution-based UEs due to their visual concealment as shown in Fig. 1 (a). We observe that the edge pixel values of the convolution-based samples are biased towards black. Motivated by this key observation, we propose a detection scheme based on statistics of edge pixel values, successfully identifying convolution-based samples from UEs. Subsequently, to simplify the challenging problem of defending against convolution-based UEs, we align with (Javanmard and Soltanolkotabi 2022; Min, Chen, and Karbasi 2021) in regarding the image samples as column vectors gener-

*These authors contributed equally.
†Corresponding author.

ated by a *Gaussian mixture model* (GMM) (Reynolds 2009). In this manner, existing convolution-based UEs can be expressed as the product of a matrix and clean samples. This multiplicative matrix can be directly understood as *multiplicative perturbations*, in contrast to the *additive perturbations* from the bounded UEs. Considering that the reason behind the effectiveness of convolution-based UEs is that DNNs incorrectly establish the mapping between class-wise multiplicative noise and the ground truth labels (Sadasivan, Soltanolkotabi, and Feizi 2023). In light of this, our key intuition is to disrupt the distribution of class-wise multiplicative noise by increasing the inconsistency within classes and enhancing the consistency between classes.

Hence, we formally define two quantitative metrics in GMM, the inconsistency within intra-class multiplicative matrix (Θ_{imi}) and the consistency within inter-class multiplicative matrix (Θ_{imc}). Our defense goal is to design an operation that can simultaneously enhance both Θ_{imi} and Θ_{imc}. Specifically, we leverage a uniform distribution to generate random values and shifts to construct a random matrix $\mathcal{A}_r$, subsequently applied to multiply the convolution-based UEs in the GMM scenario. After conducting empirical experiments, we verify that the random matrix we design effectively boosts both Θ_{imi} and Θ_{imc}, demonstrating defensive capabilities against convolution-based UEs.

Based on these insights, we propose the first defense for countering convolution-based UEs, termed as **COIN**, which employs a randomly multiplicative image transformation as its mechanism. To verify the defense generalization of COIN against convolution-based UEs, we further propose two convolution-based UEs, referred to as HUDA and VUDA as shown in Fig. 1. Extensive experiments reveal that our proposed defense COIN significantly overwhelms existing defense schemes, ranging from 19.17%-44.63% in accuracy on CIFAR-10 and CIFAR-100, while ACC exceeds 85% and AUC surpasses 0.85 of proposed detection approach EPD. Our main contributions are summarized as:

- **First Defense Against Convolution-based UEs.** To the best of our knowledge, we propose the first detection and defense for convolution-based UEs, which utilizes an edge pixel detection and a random matrix multiplication.
- **Novel Convolution-based UEs.** We newly propose two convolution-based UEs, termed as VUDA and HUDA, to supply the scope of convolution-based UEs for better convincing defense evaluation.
- **Experimental Evaluations.** Extensive experiments against convolution-based UEs including existing ones and those we newly propose on four benchmark datasets and six types of model architectures validate the effectiveness of our detection and defense strategy.

Preliminaries

Threat Model

The attacker crafts convolution-based UEs by employing class-wise convolutional kernel $\mathcal{K}_i$ to each image x_i in the training set $\mathcal{D}_c$, thus causing the model F with parameter θ trained on this dataset to generalize poorly to a clean test distribution $\mathcal{D}$ (Sadasivan, Soltanolkotabi, and Feizi 2023). Therefore, the attacker formally expects to work out the following bi-level objective:

$$\max \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} \left[\mathcal{L}\left(F\left(x;\theta_p\right),y\right)\right] \tag{1}$$

$$\textit{s.t. } \theta_p = \arg\min_{\theta} \sum_{(x_i,y_i)\in\mathcal{D}_c} \mathcal{L}\left(F\left(x_i \otimes \mathcal{K}_i;\theta\right),y_i\right) \tag{2}$$

where (x_i, y_i) represents the clean data from $\mathcal{D}_c$, $\mathcal{L}$ is a loss function, *e.g.*, cross-entropy loss, and $\otimes$ represents the convolution operation, while ensuring the modifications to x_i are not excessive for preserving the concealment of the sample. As for defenders, in the absence of any knowledge of clean samples x_i, they aim to perform certain operations on UEs to achieve the opposite goal of Eq. (1). Bounded UEs (Huang et al. 2021; Yu et al. 2022) add norm-constrained additive noise to the sample x_i in Eq. (2).

Broadening the Attack Range

Given the scarcity of research on existing convolution-based UEs, for more convincingly and comprehensively evaluating existing defense approaches against convolution-based UE attacks, we propose two types of convolution-based UEs, namely HUDA and VUDA, to expand the scope of convolution-based UEs. In particular, HUDA utilizes class-wise horizontal filters to apply convolutional operations, subsequently employed on images based on categories. The definition of HUDA's convolutional kernel is as follows:

$$\mathcal{K}_h(i,j,k,l) = \begin{cases} b_y & \text{if } j = \frac{\mathcal{T}}{2} \quad and \quad i \in \{0,1,2\} \\ 0 & \text{else} \end{cases} \tag{3}$$

where i represents the channel, j and k denote the rows and columns of the convolutional kernel, and l represents the depth of the convolutional kernel, $\mathcal{T}$ denotes the kernel size, and b_y denotes the class-wise blur parameter. Similarly, the definition of the vertical convolutional kernel of VUDA is:

$$\mathcal{K}_v(i,j,k,l) = \begin{cases} b_y & \text{if } k = \frac{\mathcal{T}}{2} \quad and \quad i \in \{0,1,2\} \\ 0 & \text{else} \end{cases} \tag{4}$$

After obtaining the class-wise convolutional kernels, the entire dataset is processed using convolution operations to add the convolution-based multiplicative perturbations.

Low-dimensional Representation

Decomposing in low-dimensional space. Similar to (Javanmard and Soltanolkotabi 2022; Min, Chen, and Karbasi 2021; Wang et al. 2024b), we define a binary classification problem involving a Bayesian classifier (Friedman, Geiger, and Goldszmidt 1997), and the clean dataset $\mathcal{D}_c$ is sampled from a Gaussian mixture model $\mathcal{N}(y\boldsymbol{\mu}, I)$. Here, y represents the labels $\{\pm 1\}$, with mean $\boldsymbol{\mu} \in \mathbb{R}^d$, and covariance $I \in \mathbb{R}^{d\times d}$ as the identity matrix (d represents the feature dimension). We denote the clean sample as $x \in \mathbb{R}^d$, the convolution-based UE as x_u, which can be formulated as left-multiplying the class-wise matrices $\mathcal{A}_y$ by x:

$$x_u = \mathcal{A}_y \cdot x \tag{5}$$

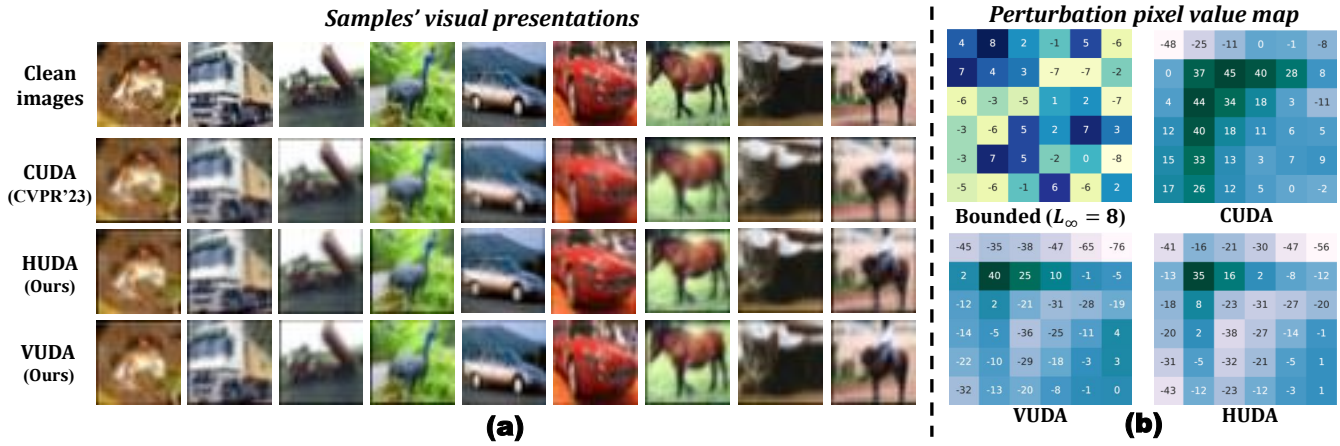

Figure 1: (a) Visual images of clean samples and three convolution-based UEs, CUDA (Sadasivan, Soltanolkotabi, and Feizi 2023), HUDA, and VUDA. (b) The plots of perturbation values from bounded and convolution-based UEs. It can be seen that the bounded perturbation values are limited within a certain range, while convolution-based perturbations lack such constraints.

Specifically, CUDA employs a tridiagonal matrix $\mathcal{A}_t$, characterized by diagonal elements equal to 1, with the lower diagonal elements and upper diagonal elements both set to a pre-defined parameter a_y.

Key intuition. The reason why convolution-based UEs work is that the model establishes a mapping between class-wise multiplicative noise and labels (Sadasivan, Soltanolkotabi, and Feizi 2023). Therefore, our intuition is that converting class-wise noise into random noise can disrupt this mapping. Specifically, by increasing the inconsistency of intra-class multiplicative matrices or the consistency of inter-class multiplicative matrices, we can make the class-wise multiplicative noise more disordered, resulting in models being unable to learn meaningless noise. Hence, we define the objective in the GMM as follows:

Definition 1. *We define **intra-class matrix inconsistency**, denoted as Θ_{imi}, as follows: Given the multiplicative matrices $\{\mathcal{A}_i \mid i = 1, 2, \cdots, n\}$ within a certain class y_k (containing n samples), we have an intra-class average matrix defined as $\mathcal{A}_{\mu_k} = \frac{1}{n}\sum_{i=1}^{n}\mathcal{A}_i$, an intra-class matrix variance defined as $\mathbb{D}_k = \frac{1}{n}\sum_{i=1}^{n}(\mathcal{A}_i - \mathcal{A}_{\mu_k})^2 \in \mathbb{R}^{d\times d}$, an intra-class matrix variance mean value defined as $\mathcal{V}_{mk} = \frac{1}{d^2}\sum_{i=0}^{d-1}\sum_{j=0}^{d-1}\mathbb{D}_k[i][j]$, and we have $\Theta_{imi} = \frac{1}{c}\sum_{k=0}^{c-1}\mathcal{V}_{mk}$, where c denotes the number of classes in $\mathcal{D}_c$.*

Definition 2. *We define **inter-class matrix consistency** as Θ_{imc}, as follows: Given the flattened intra-class average matrices of the j-th and k-th classes $flat(\mathcal{A}_{\mu_j})$, $flat(\mathcal{A}_{\mu_k})$, we have $\Theta_{imc} = sim(flat(\mathcal{A}_{\mu_j}), flat(\mathcal{A}_{\mu_k}))$, where $flat()$ denotes flattening the matrix into a row vector and $sim(u, v) = uv^T/(\|u\|\,\|v\|)$ denotes cosine similarity.*

Hypothesis. *Increasing Θ_{imi} or Θ_{imc} both improves the test accuracy of classifiers trained on the convolution-based UEs, and vice versa.*

Empirical validation. With the two metrics, we conduct experiments to validate our key intuition. It can be observed from the top row of Fig. 2 (Θ_{imi} remains constant) that an increasing Θ_{imc} corresponds to an improvement in accuracy, whereas a decrease in Θ_{imc} leads to a decline in accuracy. In the bottom row of Fig. 2 (Θ_{imc} is constant), accuracy increases as Θ_{imi} rises and decreases as Θ_{imi} falls. Hence, these results support our proposed hypothesis.

Design of Low-dimensional Defense

Based on the results, we are motivated to design a scheme that increases Θ_{imi} and Θ_{imc}, thus perturbing the distributions of $\mathcal{A}_y$. Assuming $\mathcal{A}_y$ is the most common diagonal matrix, we left-multiply $\mathcal{A}_y$ by a random matrix $\mathcal{A}_r \in \mathbb{R}^{d\times d}$, where the diagonal of $\mathcal{A}_r$ is set to random values to introduce randomness. However, the form of diagonal matrix remains unchanged by multiplying this diagonal matrix, limiting the randomness. To solve this, we add another set of random variables above the diagonal, making it random. However, $\mathcal{A}_t$ employed in CUDA still maintains the tridiagonal pattern, still contraining the randomness. To introduce more randomness, we further introduce small random offsets to the random variables for each row, thus breaking the tridiagonal form, without introducing additional variables.

Thus, we unify the random values and shifts via a uniform distribution $\mathcal{U}(-\alpha, \alpha)$, thereby striving to enhance Θ_{imc} while already improving Θ_{imi}. We first sample a variable $s \sim \mathcal{U}(-\alpha, \alpha), s \in \mathbb{R}^d$, and then obtain $m_i = \lfloor s_i \rfloor, n_i = s_i - m_i, 0 \le i \le d-1$. $\mathcal{A}_r$ is designed as:

$$\mathcal{A}_r = \begin{bmatrix} \underline{1-n_0} & \underline{n_0} & 0 & 0 & \dots & 0 \\ 0 & \underline{1-n_1} & \underline{n_1} & 0 & \dots & 0 \\ 0 & 0 & \underline{1-n_2} & \underline{n_2} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & \dots & \dots & 0 & \underline{1-n_{d-1}} \end{bmatrix} \in \mathbb{R}^{d\times d} \tag{6}$$

where the i-th and (i+1)-th elements of the i-th row ($0 \le i \le d$-1) are $1 - n_i$ and n_i, and "$\underline{1-n_i, n_i}$" means that the positions of these two elements for each row are shifted by m_i units simultaneously. When the new location $i+m_i$ or $i+1+m_i$ exceeds the matrix boundaries, we take its modulus with respect to d, thus obtaining their new positions.

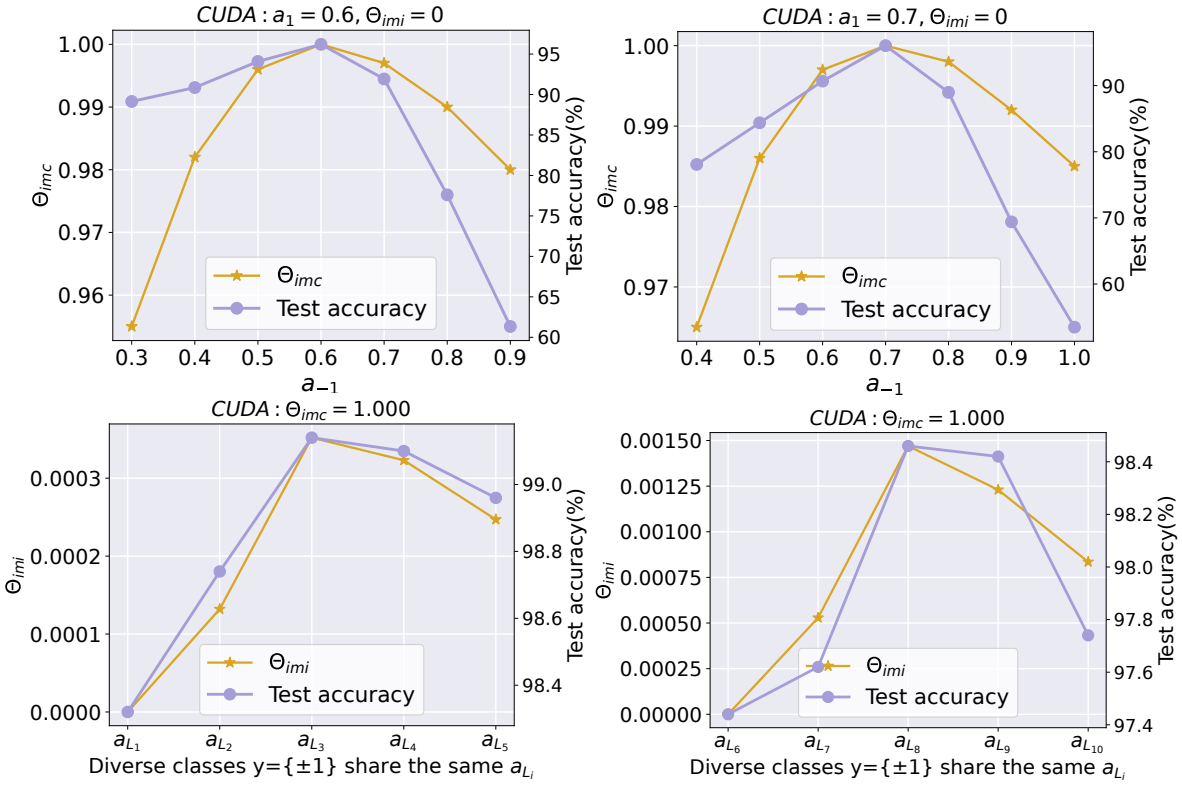

Figure 2: Hypothesis validation. Test accuracy (%) with Θ_{imc} (top row) and Θ_{imi} (bottom row) via changing parameter a_y.

Whereupon, we obtain accuracy results by left-multiplying CUDA samples with $\mathcal{A}_r$. As shown in Fig. 3, the w/ $\mathcal{A}_r$ accuracy is ahead of the w/o $\mathcal{A}_r$ accuracy regardless of a_1, indicating the effectiveness of $\mathcal{A}_r$.

Methodology

Our Design for Defense Scheme: COIN

Given the effectiveness of $\mathcal{A}_r$ in the low-dimensional space, our intuition is that extending it to the real-image domain also works against convolution-based UEs. To achieve this, we model the two random values in each row in Eq. (6) as two weighting coefficients, while exploiting the random location offset m_i in each row to locate the pixel positions. Therefore, we regard the process of multiplying $\mathcal{A}_r$ as a random linear interpolation process, *i.e.*, a random multiplicative transformation. Thus, the convolution-based UE $x_u \in \mathbb{R}^{C\times H\times W}$ requires variables along with both horizontal and vertical directions and leveraging bilinear interpolation, which are defined as follows:

$$s_x, s_y \sim \mathcal{U}(-\alpha, \alpha, size = H \cdot W) \tag{7}$$

where $\mathcal{U}$ denotes a uniform distribution with size of $H \cdot W$ (height $\times$ width), α controls the range of the random variables. Considering that s_x and s_y are both arrays with size of $H \cdot W$, we obtain arrays m_x and m_y by rounding down each variable from the arrays to its integer part (*i.e.*, random location offsets), formulated as:

$$m_{xi} = \lfloor s_{xi} \rfloor, \quad m_{y_i} = \lfloor s_{y_i} \rfloor \tag{8}$$

where $\lfloor\ \rfloor$ represents the floor function, and i is the index in the array, ranging from 0 to $H \cdot W - 1$. Subsequently, the arrays with coefficients ω_x, ω_y required for the bilinear interpolation process are computed as follows:

$$\omega_{xi} = s_{xi} - m_{xi}, \quad \omega_{y_i} = s_{xi} - m_{y_i} \tag{9}$$

To obtain the coordinates of the pixels for interpolation, we initialize a coordinate grid, defined as:

$$c_x, c_y = \mathcal{M}(ara(W), ara(H)) \in \mathbb{R}^{H\times W} \tag{10}$$

where $\mathcal{M}$ denotes coordinate grid creation function, ara is employed to produce an array with values evenly distributed within a specified range. Whereupon we obtain the coordinates of the four nearest pixel points around the desired interpolation point in the bilinear interpolation process:

$$q_{11_i} = ((c_{xi} + m_{xi})\%W, (c_{y_i} + m_{y_i})\%H) \tag{11}$$

$$q_{21_i} = ((c_{xi} + m_{xi} + 1)\%W, (c_{y_i} + m_{y_i})\%H) \tag{12}$$

$$q_{12_i} = ((c_{xi} + m_{xi})\%W, (c_{y_i} + m_{y_i} + 1)\%H) \tag{13}$$

$$q_{22_i} = ((c_{xi} + m_{xi} + 1)\%W, (c_{y_i} + m_{y_i} + 1)\%H) \tag{14}$$

where $\%$ represents the modulo function, ensuring that the horizontal coordinate ranges from 0 to $W - 1$ and the vertical coordinate ranges from 0 to $H - 1$. Hence, we obtain new pixel values by using the pixel values of the four points through bilinear interpolation:

$$\mathcal{F}_j(p_i) = \begin{bmatrix} 1-\omega_{xi} & \omega_{xi} \end{bmatrix} \begin{bmatrix} \mathcal{F}_j(q_{11_i}) & \mathcal{F}_j(q_{12_i}) \\ \mathcal{F}_j(q_{21_i}) & \mathcal{F}_j(q_{22_i}) \end{bmatrix} \begin{bmatrix} 1-\omega_{y_i} \\ \omega_{y_i} \end{bmatrix} \tag{15}$$

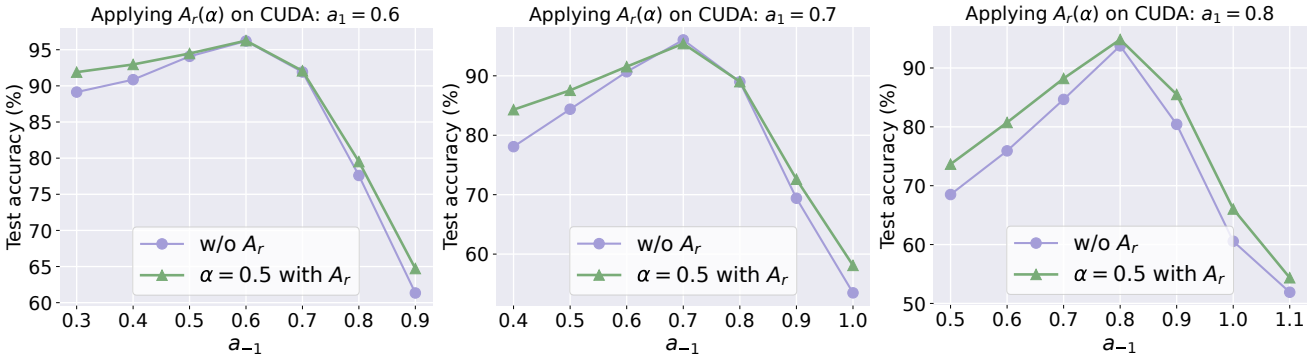

Figure 3: Comparison results of test accuracy before and after left-multiplying $\mathcal{A}_r$ on CUDA samples, α is set to 0.5.

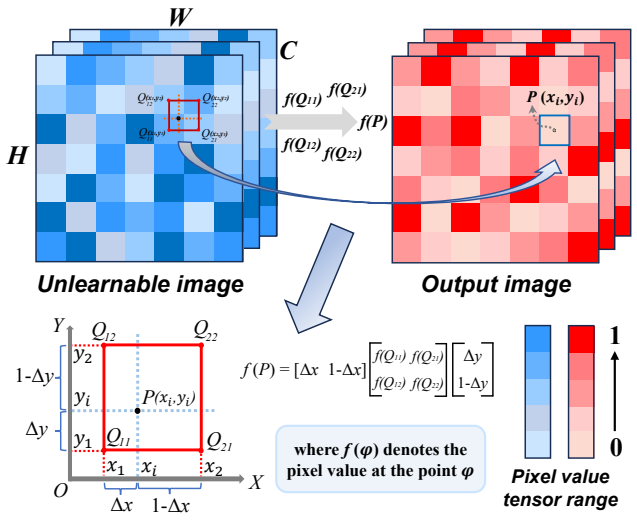

Figure 4: Our defense scheme COIN

where $\mathcal{F}_j()$ denotes the value of the j-th channel at a certain coordinate point, p_i denotes the coordinate of the newly generated pixel point. Finally, we gain the transformed image x_t by applying Eq. (15) and clipping to pixel values of each channel of x_u. The pipeline of COIN is in Fig. 4.

Our Design for Detection Scheme: EPD

Key Intuition. It is crucial to detect whether the perturbations are convolution-based or not, because without prior knowledge, defenders have difficulty discerning the type of UEs. The intuition behind our detection method is that convolution-based UEs typically exhibit lower pixel values around the edges, often appearing as black pixels as shown in Fig. 1 (a). Leveraging this, we propose *edge pixel-based detection* (EPD) scheme as follows:

Feature Extraction. We sum the RGB values along the four edges (*i.e.*, top, bottom, left, and right) of the detection-ready sample $x_d \in \mathbb{R}^{C\times H\times W}$ to serve as the feature for detection. The channel-wise feature vector $v_d(c)$ is defined as:

$$v_d(c) = \left[\sum_{w=1}^{W} e(c,1,w), \sum_{w=1}^{W} e(c,H,w), \sum_{h=1}^{H} e(c,h,1), \sum_{h=1}^{H} e(c,h,W)\right] \tag{16}$$

where $e(c,h,w)$ represents the pixel value at channel c, row h, and column w. The final feature vector $\mathcal{V}_d$ is calculated by concatenating $v_d(c)$ across three channels, formulated as:

$$\mathcal{V}_d = [v_d(R), v_d(G), v_d(B)] \tag{17}$$

Binary Classification for Detection. Given the simplicity of the vector $\mathcal{V}_d$, we utilize a *support vector machine* (SVM) (Huang et al. 2018) for binary classification. The pre-trained SVM classifies the image x_d by comparing the decision score $\mathcal{S}$ with a pre-defined threshold θ (typically set to 0 in SVM), where $\mathcal{S}$ is defined as:

$$\mathcal{S} = \mathbf{w}\cdot\mathcal{V}_d + b \tag{18}$$

where $\mathbf{w}$ is the weight vector of the SVM, and b is the bias term. The predicted label $\hat{y} \in \{0,1\}$ is given by:

$$\hat{y} = \begin{cases} 1 & \text{if } S(x_d) \geq \theta \\ 0 & \text{if } S(x_d) < \theta \end{cases} \tag{19}$$

where 1 denotes a convolution-based UE, while 0 does not. Since our defense COIN is designed specifically for convolutional UEs, we only apply COIN after our EPD identifies the sample as a convolutional UE. Otherwise, whether the sample is a bounded UE, a clean example, or any other type, we do not consider differentiation or processing.

Experiments

Experimental Settings

Networks including ResNet (RN) (He et al. 2016), DenseNet (DN) (Huang et al. 2017), MobileNetV2 (MNV2) (Sandler et al. 2018), and VGG (Simonyan and Zisserman 2014) are selected. Benchmark datasets including CIFAR-10 (32×32), CIFAR-100 (32×32) (Krizhevsky and Hinton 2009), and ImageNet (224×224) (Deng et al. 2009) are used. The uniform distribution range α of COIN is empirically set to 2.0. During pre-training the SVM classifier of EPD, the linear kernel function is utilized, and penalty parameter C_p is set to 0.1. Meanwhile, the threshold θ of EPD during detection is set

Datasets	CIFAR-10					CIFAR-100				
Defenses	ResNet18	VGG16	DenseNet121	MobileNetV2	**AVG**	ResNet18	VGG16	DenseNet121	MobileNetV2	**AVG**
w/o	26.49	24.65	27.21	21.34	24.92	14.31	12.53	13.90	12.94	13.42
MU	26.72	28.07	24.67	24.63	26.02	17.09	13.35	19.97	13.55	15.99
CM	26.02	28.53	24.64	20.73	24.98	12.51	10.14	20.77	10.14	13.39
CO	20.07	27.58	24.86	20.46	23.24	12.80	10.56	16.19	13.56	13.28
DP-SGD	25.50	23.02	25.25	25.78	24.89	12.42	10.56	16.36	12.72	13.02
AVATAR	30.67	29.57	33.15	28.53	30.48	14.49	10.81	12.97	13.85	13.03
AA	39.85	38.68	38.92	41.06	39.63	24.83	1.00	27.89	20.49	18.55
OP	29.77	30.33	33.82	28.86	30.70	20.17	14.59	15.55	23.02	18.33
ISS-G	25.77	21.42	26.73	19.85	23.44	8.80	6.40	11.48	8.71	8.85
ISS-J	45.10	40.26	39.79	41.46	41.65	33.62	26.92	28.94	31.23	30.18
AT	50.59	45.95	49.01	42.59	47.04	37.27	28.18	34.21	35.74	33.85
ECLIPSE	32.87	30.82	32.26	32.69	32.16	18.54	15.59	19.24	20.23	18.40
COIN (Ours)	**71.90**	**73.65**	**70.45**	**73.63**	**72.41**	**48.63**	**46.74**	**45.72**	**48.53**	**47.41**

Table 2: Defenses against CUDA. The *test accuracy* (%) results on CIFAR-10 and CIFAR-100 datasets.

Dataset	ImageNet100 224×224				
Defense	RN18	RN50	DN121	MNV2	**AVG**
w/o	25.74	26.66	21.70	16.30	22.60
CO	25.46	29.20	23.90	17.58	24.04
MU	34.96	19.38	27.78	15.60	24.43
CM	16.54	24.04	23.58	8.00	18.04
ISS-G	14.92	13.50	9.78	5.78	11.00
AT	**37.82**	36.80	30.34	41.42	36.60
ISS-J	30.10	**37.04**	25.52	28.04	30.18
COIN	37.80	35.38	**35.22**	**41.50**	**37.48**

Table 3: Defenses against CUDA on ImageNet100.

to 0. During training of classifiers, we use SGD for training 80 epochs with a momentum of 0.9, a learning rate of 0.1, and batch sizes of 128, 32 for CIFAR and ImageNet, respectively. Each result is run once on a 3090 GPU.

Comparison with SOTA Defenses

We compare COIN with 10 SOTA defense schemes, *i.e.*, AT (Tao et al. 2021), ISS (Liu, Zhao, and Larson 2023) including JPEG compression (ISS-J), and grayscale transformation (ISS-G), AVATAR (Dolatabadi, Erfani, and Leckie 2023), OP (Segura et al. 2023), AA (Qin et al. 2023), and ECLIPSE (Wang et al. 2024a). We also apply four defenses, differential privacy SGD (DP-SGD) (Hong et al. 2020; Zhang et al. 2021), cutmix (CM) (Yun et al. 2019), cutout (CO) (DeVries and Taylor 2017), and mixup (MU) (Zhang et al. 2018) that are popularly used to test against UEs.

Evaluation Metrics

Consistent with previous works (Wang et al. 2024a,c), we employ $test\ accuracy$, *i.e.*, the model accuracy on a clean test set after training on the datasets. The higher the $test\ accuracy$, the better the defense effectiveness. For evaluation of the detection performance, similar to Liu et al. (2023), we employ *binary classification accuracy* (ACC) (%) and *Area Under the Receiver Operating Characteristic Curve* (AUC) to measure the performance, where AUC is the area under the ROC curve, indicating the model's ability to distinguish between classes. Ranging from 0 to 1, a higher AUC signifies better detection performance.

Evaluation of Defense Scheme COIN

The accuracy (%) results are shown in Tabs. 2 to 4. The values covered by gray denote the best effect. It can be observed the average performance of existing defenses lag behind COIN , ranging from 19.17% to 44.63% as shown in Tab. 2. Meanwhile, COIN maintains an advantage of 0.88%-26.48% in Tab. 3. The reason AT lags behind COIN can be deduced from that the multiplicative noise underlying convolutional UEs is more easily defeated by the multiplicative random noise used by COIN, rather than the additive noise on which AT is based (Madry et al. 2018). We also evaluate defenses against VUDA and HUDA in Tab. 4. It can be seen that COIN is effective in defending against these convolutional UEs and outperforms defenses by 23.84% to 40%. Regarding practicality, while we acknowledge that COIN cannot be effective for all UEs, it is still effective against some bounded UEs like URP (Tao et al. 2021) and OPS (Wu et al. 2023), which improves accuracy from 16.8%, 28.4% to 81.1%, 80.1%.

Evaluation of Detection Scheme EPD

We formulate the detection scenario with a dataset (size of 50000) combining an even distribution of six UEs among 12 SOTA UEs CUDA (Sadasivan, Soltanolkotabi, and Feizi 2023), HUDA, VUDA, OPS (Wu et al. 2023), AR (Sandoval-Segura et al. 2022), URP (Tao et al. 2021), UHP (Tao et al. 2021), LSP (Yu et al. 2022), TAP (Fowl et al. 2021), EM (Huang et al. 2021), EFP (Wen et al. 2023), and SEP (Chen et al. 2023). We set the following combinations for CIFAR-10:

- $\mathcal{S}_1 \longrightarrow$ CUDA, VUDA, HUDA, OPS, AR, URP
- $\mathcal{S}_2 \longrightarrow$ CUDA, VUDA, HUDA, OPS, LSP, URP
- $\mathcal{S}_3 \longrightarrow$ CUDA, VUDA, HUDA, TAP, EM, EFP
- $\mathcal{S}_4 \longrightarrow$ CUDA, VUDA, HUDA, OPS, EFP, SEP,

and the following is for ImageNet-20:

- $\mathcal{S}_5 \longrightarrow$ CUDA, VUDA, HUDA, LSP, AR, URP

Dataset	VUDA CIFAR-10					HUDA CIFAR-10				
Defenses	RN18	VGG16	DN121	MNV2	**AVG**	RN18	VGG16	DN121	MNV2	**AVG**
w/o	40.36	43.95	44.03	36.86	41.30	39.65	36.63	50.32	34.73	40.53
AVATAR	39.25	38.68	41.64	39.37	39.74	43.53	39.84	52.10	46.28	42.27
AA	51.87	10.00	60.09	51.42	43.35	56.83	10.00	60.09	61.08	46.27
OP	42.67	39.26	46.25	41.04	42.31	45.36	46.77	59.80	51.79	49.21
ISS-G	38.50	32.88	38.46	33.66	35.88	22.43	43.08	29.33	34.52	33.05
ISS-J	51.49	45.00	45.54	38.57	45.15	49.89	49.48	42.23	40.58	45.47
AT	43.40	38.68	46.88	36.33	41.32	48.62	41.19	50.48	43.74	45.07
ECLIPSE	31.09	33.47	35.56	29.86	32.49	34.57	39.04	41.12	35.35	37.52
COIN	**74.98**	**68.74**	**74.96**	**71.07**	**72.44**	**75.18**	**71.75**	**75.78**	**70.10**	**73.05**

Table 4: Defenses against VUDA and HUDA. The *test accuracy* (%) results on CIFAR-10 dataset.

Dataset	CIFAR-10 UEs				ImageNet20 UEs		
$\mathcal{S}$	$\mathcal{S}_1$	$\mathcal{S}_2$	$\mathcal{S}_3$	$\mathcal{S}_4$	$\mathcal{S}_5$	$\mathcal{S}_6$	$\mathcal{S}_7$
ACC	90.40	90.94	87.89	87.63	99.92	99.93	99.89
AUC	0.904	0.909	0.878	0.876	0.999	0.999	0.998

Table 5: Evaluation of EPD. ACC (%) and AUC results.

- $\mathcal{S}_6 \longrightarrow$ CUDA, VUDA, HUDA, EM, TAP, UHP
- $\mathcal{S}_7 \longrightarrow$ CUDA, VUDA, HUDA, TAP, UHP, AR.

The results of EPD on CIFAR-10 and ImageNet20 are shown in Tab. 5. It can be seen that EPD's ACC and AUC are both high, demonstrating excellent detection performance.

Hyperparameter Analysis

We investigate the impact of α on COIN in Fig. 5. The test accuracy against CUDA increases initially with the rise in α and then decreases. This is because initially, as α increases, the CUDA perturbations gradually become disrupted. However, as α continues to increase, excessive corruptions damage image features, compromising defense. We empirically opt for an α of 2.0 that yields the highest average effect.

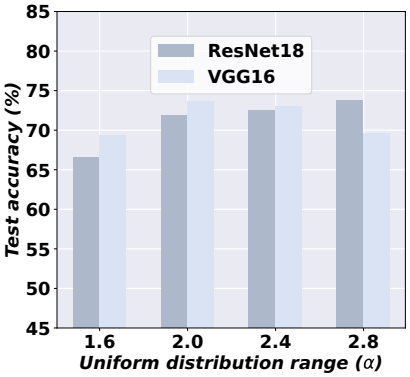

Figure 5: The impact of α on COIN.

Related Work

Unlearnable Examples

Current UEs are classified into bounded UEs and convolution-based UEs. The bounded UEs utilize shortcut-based additive unlearnable noise (Huang et al. 2021; Fowl et al. 2021; Fu et al. 2022; Wu et al. 2023). Nonetheless, recent studies indicate that simple defenses like JPEG compression addresses these bounded UEs (Liu, Zhao, and Larson 2023). Recently, Sadasivan, Soltanolkotabi, and Feizi propose a type of convolution-based UE via class-wise filters to generate convolutional noise without norm constraints. Unfortunately, current defenses prove ineffective against it, and no research has explored viable defenses. To expand the scope of convolution-based UEs, we propose two types of convolution-based UEs via vertical and horizontal filters, termed as VUDA and HUDA, revealing that these convolution-based UEs easily defeat existing defenses.

Defenses Against UEs

Tao et al. (2021) indicate AT (Madry et al. 2018) effectively defends against bounded UEs, and Liu et al. (2023) discover even JPEG compression effectively defends against UEs. Thereafter, Qin et al. (2023) employ *adversarial augmentations* (AA), and Wang et al. (2024a) (ECLIPSE) purify UEs via diffusion models (Ho, Jain, and Abbeel 2020), Segura et al. (2023) train a linear regression model to perform *orthogonal projection* (OP), while Yu et al. (2024) corrupt UEs via rate-constrained variational auto-encoders. Nevertheless, none of them are tailor-made for convolution-based UEs. For detection, although Zhu et al. (2024) propose a method for UEs, it is limited to bounded UEs and does not consider convolution types.

Conclusion

In this work, we reveal that existing defenses are ineffective against convolutional UEs. In response, we propose an edge-pixel detection scheme to identify convolutional samples. Besides, we demonstrate that multiplying a random matrix mitigates convolution-based UEs. Based on this, we propose the first effective defense against convolution-based UEs via pixel-interpolation random multiplication. We further propose two convolution-based UEs, VUDA and HUDA, to expand the scope of convolutional UEs. Extensive experiments verify the effectiveness of our proposed defenses.

Acknowledgements

Minghui's work is supported by the National Natural Science Foundation of China (Grant No. 62202186). Shengshan's work is supported by the National Natural Science Foundation of China (Grant No. 62372196). Xianlong Wang is the corresponding author.

References

Chen, S.; Yuan, G.; Cheng, X.; Gong, Y.; Qin, M.; Wang, Y.; and Huang, X. 2023. Self-ensemble protection: Training checkpoints are good data protectors. In *Proceedings of the 11th International Conference on Learning Representations (ICLR'23)*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *Proceedings of the 2009 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'09)*, 248–255.

DeVries, T.; and Taylor, G. W. 2017. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*.

Dolatabadi, H. M.; Erfani, S.; and Leckie, C. 2023. The devil's advocate: Shattering the illusion of unexploitable data using diffusion models. *arXiv preprint arXiv:2303.08500*.

Fowl, L.; Goldblum, M.; Chiang, P.-y.; Geiping, J.; Czaja, W.; and Goldstein, T. 2021. Adversarial examples make strong poisons. In *Proceedings of the 35th Neural Information Processing Systems (NeurIPS'21)*, volume 34, 30339–30351.

Friedman, N.; Geiger, D.; and Goldszmidt, M. 1997. Bayesian network classifiers. *Machine Learning*, 29: 131–163.

Fu, S.; He, F.; Liu, Y.; Shen, L.; and Tao, D. 2022. Robust unlearnable examples: Protecting data privacy against adversarial learning. In *Proceedings of the 10th International Conference on Learning Representations (ICLR'22)*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the 2016 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'16)*, 770–778.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. In *Proceedings of the 34th Neural Information Processing Systems (NeurIPS'20)*, volume 33, 6840–6851.

Hong, S.; Chandrasekaran, V.; Kaya, Y.; Dumitraş, T.; and Papernot, N. 2020. On the effectiveness of mitigating data poisoning attacks with gradient shaping. *arXiv preprint arXiv:2002.11497*.

Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Densely connected convolutional networks. In *Proceedings of the 2017 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'17)*, 4700–4708.

Huang, H.; Ma, X.; Erfani, S. M.; Bailey, J.; and Wang, Y. 2021. Unlearnable examples: Making personal data unexploitable. In *Proceedings of the 9th International Conference on Learning Representations (ICLR'21)*.

Huang, S.; Cai, N.; Pacheco, P. P.; Narrandes, S.; Wang, Y.; and Xu, W. 2018. Applications of support vector machine learning in cancer genomics. *Cancer genomics & proteomics*, 15(1): 41–51.

Javanmard, A.; and Soltanolkotabi, M. 2022. Precise statistical analysis of classification accuracies for adversarial training. *The Annals of Statistics*, 50(4): 2127–2156.

Krizhevsky, A.; and Hinton, G. 2009. Learning multiple layers of features from tiny images.

Liu, S.; Wang, Y.; and Gao, X.-S. 2024. Game-theoretic unlearnable example generator. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI'24)*, volume 38, 21349–21358.

Liu, X.; Li, M.; Wang, H.; Hu, S.; Ye, D.; Jin, H.; Wu, L.; and Xiao, C. 2023. Detecting Backdoors During the Inference Stage Based on Corruption Robustness Consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'23)*, 16363–16372.

Liu, Z.; Zhao, Z.; and Larson, M. 2023. Image shortcut squeezing: Countering perturbative availability poisons with compression. In *Proceedings of the 40th International Conference on Machine Learning (ICML'23)*.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards deep learning models resistant to adversarial attacks. In *Proceedings of the 6th International Conference on Learning Representations (ICLR'18)*.

Meng, R.; Yi, C.; Yu, Y.; Yang, S.; Shen, B.; and Kot, A. C. 2024. Semantic deep hiding for robust unlearnable examples. *IEEE Transactions on Information Forensics and Security (TIFS'24)*.

Min, Y.; Chen, L.; and Karbasi, A. 2021. The curious case of adversarially robust models: More data can help, double descend, or hurt generalization. In *Uncertainty in Artificial Intelligence*, 129–139. PMLR.

Qin, T.; Gao, X.; Zhao, J.; Ye, K.; and Xu, C.-Z. 2023. Learning the unlearnable: Adversarial augmentations suppress unlearnable example attacks. *arXiv preprint arXiv:2303.15127*.

Reynolds, D. A. 2009. Gaussian mixture models. *Encyclopedia of Biometrics*, 741(659-663).

Sadasivan, V. S.; Soltanolkotabi, M.; and Feizi, S. 2023. Cuda: Convolution-based unlearnable datasets. In *Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'23)*, 3862–3871.

Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; and Chen, L.-C. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'18)*, 4510–4520.

Sandoval-Segura, P.; Singla, V.; Geiping, J.; Goldblum, M.; Goldstein, T.; and Jacobs, D. W. 2022. Autoregressive perturbations for data poisoning. In *Proceedings of the 36th Neural Information Processing Systems (NeurIPS'22)*, volume 35.

Segura, P. S.; Singla, V.; Geiping, J.; Goldblum, M.; and Goldstein, T. 2023. What can we learn from unlearnable datasets? In *Proceedings of the 37th Neural Information Processing Systems (NeurIPS'23)*.

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Tao, L.; Feng, L.; Yi, J.; Huang, S.-J.; and Chen, S. 2021. Better safe than sorry: Preventing delusive adversaries with adversarial training. In *Proceedings of the 35th Neural Information Processing Systems (NeurIPS'21)*, volume 34, 16209–16225.

Wang, X.; Hu, S.; Zhang, Y.; Zhou, Z.; Zhang, L. Y.; Xu, P.; Wan, W.; and Jin, H. 2024a. ECLIPSE: Expunging clean-label indiscriminate poisons via sparse diffusion purification. In *Proceedings of the 29th European Symposium on Research in Computer Security (ESORICS'24)*.

Wang, X.; Li, M.; Liu, W.; Zhang, H.; Hu, S.; Zhang, Y.; Zhou, Z.; and Jin, H. 2024b. Unlearnable 3D point clouds: Class-wise transformation is all you need. In *Proceedings of the 38th Conference on Neural Information Processing Systems (NeurIPS'24)*.

Wang, X.; Li, M.; Xu, P.; Liu, W.; Zhang, L. Y.; Hu, S.; and Zhang, Y. 2024c. PointAPA: Towards availability poisoning attacks in 3D point clouds. In *Proceedings of the European Symposium on Research in Computer Security (ESORICS'24)*, 125–145. Springer.

Wen, R.; Zhao, Z.; Liu, Z.; Backes, M.; Wang, T.; and Zhang, Y. 2023. Is adversarial training really a silver bullet for mitigating data poisoning? In *Proceedings of the 11th International Conference on Learning Representations (ICLR'23)*.

Wu, S.; Chen, S.; Xie, C.; and Huang, X. 2023. One-pixel shortcut: On the learning preference of deep neural networks. In *Proceedings of the 11th International Conference on Learning Representations (ICLR'23)*.

Yu, D.; Zhang, H.; Chen, W.; Yin, J.; and Liu, T.-Y. 2022. Availability attacks create shortcuts. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'22)*, 2367–2376.

Yu, Y.; Wang, Y.; Xia, S.; Yang, W.; Lu, S.; Tan, Y.-P.; and Kot, A. C. 2024. Purify unlearnable examples via rate-constrained variational autoencoders. In *Proceedings of the 41th International Conference on Machine Learning (ICML'24)*.

Yun, S.; Han, D.; Oh, S. J.; Chun, S.; Choe, J.; and Yoo, Y. 2019. CutMix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the 17th International Conference on Computer Vision (ICCV'19)*.

Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2018. Mixup: Beyond empirical risk minimization. In *Proceedings of the 6th International Conference on Learning Representations (ICLR'18)*.

Zhang, L.; Shen, B.; Barnawi, A.; Xi, S.; Kumar, N.; and Wu, Y. 2021. FedDPGAN: Federated differentially private generative adversarial networks framework for the detection of COVID-19 pneumonia. *Information Systems Frontiers*, 23(6): 1403–1415.

Zhu, Y.; Yu, L.; and Gao, X.-S. 2024. Detection and defense of unlearnable examples. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI'24)*, volume 38, 17211–17219.