TRANSFERABLE UNLEARNABLE EXAMPLES

Jie Ren[*]
Michigan State University
renjie3@msu.edu

Han Xu[*]
Michigan State University
xuhan1@msu.edu

Yuxuan Wan
Michigan State University
wanyuxua@msu.edu

Xingjun Ma
Fudan University
xingjunma@fudan.edu.cn

Lichao Sun
Lehigh University
lis221@lehigh.edu

Jiliang Tang
Michigan State University
tangjili@msu.edu

ABSTRACT

With more people publishing their personal data online, unauthorized data usage has become a serious concern. The *unlearnable* strategies have been introduced to prevent third parties from training on the data without permission. They add perturbations to the users' data before publishing, which aims to make the models trained on the perturbed published dataset invalidated. These perturbations have been generated for a specific training setting and a target dataset. However, their unlearnable effects significantly decrease when used in other training settings and datasets. To tackle this issue, we propose a novel unlearnable strategy based on *Classwise Separability Discriminant* (CSD), which aims to better transfer the unlearnable effects to other training settings and datasets by enhancing the linear separability. Extensive experiments demonstrate the transferability of the proposed unlearnable examples across training settings and datasets.

1 INTRODUCTION

With more people publishing their personal data online, it has raised the concern that the published data can be utilized without the data owner's permission to train machine learning models unauthorizedly. Large-scale datasets collected from Internet like LFW(Huang et al., 2008), Freebase(Bollacker et al., 2008), and Ms-celeb-1m(Guo et al., 2016) promote the development of deep learning. However, they also have the potential risk of privacy leakage. Thus, growing efforts (Huang et al., 2020; Fowl et al., 2021) have been made on protecting data from unauthorized usage by making the data samples unlearnable (Huang et al., 2020; Fowl et al., 2021; He et al., 2022). In these methods, they generate the unlearnable examples by injecting imperceptible "shortcut" perturbation. If the data is used by unauthorized training, the models prefer to extract such easy-to-learn shortcut features and ignore the semantic information in the original data (Geirhos et al., 2020). Thus, the trained model fails to recognize the user's data during the test phase and the user's data gets protected.

However, the majority of existing methods have weaknesses in two types of transferability, training-wise and data-wise, which can largely limit the practical use of unlearnable examples. **First,** weak training-wise transferability implies that the perturbed samples generated towards one target training setting, such as ERM, cannot be transferred to other training settings. As shown in Section 3.2.1, although Error-Minimizing Noise (EMN) (Huang et al., 2020) can protect data from supervised training, we can use unsupervised learning methods, such as Contrastive Learning (Chen et al., 2020a; Chen & He, 2021; Chen et al., 2020b), to first learn useful representations from the EMN-perturbed dataset, and then fine-tune the model on the perturbed dataset. In this way, the unsupervised model can also achieve comparable performance with the model that is trained on the unperturbed dataset. **Second**, insufficient data-wise transferability indicates that the unlearnable effect of perturbations generated for one target dataset will significantly decrease when transferred to other datasets. Without effective data-wise transferability, we have to generate perturbations for each dataset, which makes the unlearnable process inflexible in reality. For example, data in various applications such as social media is dynamic or even streaming and it is challenging to generate the entire perturbation set when new data is continuously emerging.

[*] Equal contribution.

In this work, we aim to enhance the training-wise and data-wise transferability of unlearnable examples. In detail, our method is motivated by the method Synthetic Noise (SN) (Yu et al., 2021), which devises a manually designed linear separable perturbation to generate unlearnable examples. Such perturbation does not target specific dataset, thus it has the potential to enhance data-wise transferability. However, SN is manually designed and it is not quantifiable or optimizable. Therefore, it is impossible to incorporate SN into other optimization processes. Meanwhile, SN lacks training-wise transferability. Therefore, in our paper, we propose *Classwise Separability Discriminant* (CSD) to generate optimizable linear-separable perturbations. Our framework *Transferable Unlearnable Examples* with enhanced linear separability can generate unlearnable examples with superior training-wise and data-wise transferability.

2 Related Work

Unlearnable Examples. Unlearnable examples are close to availability attack which aims at making the data out of service for training the models by the third parties (Muñoz-González et al., 2017). Before releasing the data in public, we can make it unlearnable to stop others from using it for training ML model without permission. Several works produce unlearnable examples with the guidance of the label information (Huang et al., 2020; Fowl et al., 2021; Shan et al., 2020; Yu et al., 2021). The vanilla unlearnable examples are produced by an alternating bi-level min-min optimization on both model parameters and perturbations (Huang et al., 2020). Being induced to trust that the perturbation can minimize the loss better than the original image features, the model will pay more attention to the perturbations. Yu et al. (2021) pointed out that a common property behind the perturbations is linear separability in input space. We refer to unlearnable examples generated under the supervised setting as **supervised unlearnable examples**. Very recently, Unlearnable Contrastive Learning (UCL) is proposed He et al. (2022) to generate unlearnable examples based on unsupervised contrastive learning and to protect data from unsupervised learning. We refer to unlearnable examples generated under the unsupervised setting as **unsupervised unlearnable examples**. Our studies show that both supervised and unsupervised unlearnable examples lack training-wise transferability as shown in Section 3.2.1 and Appendix A, respectively. Table 1 summarizes the settings of existing methods, where ✗ means ineffective protection.

Table 1: Unlearnable Examples Settings of Existing Approaches

Prevented unauthorized use	Supervised Training	Unsupervised Training
Supervised Unlearnability	Huang et al. (2020); Fowl et al. (2021)	✗
Unsupervised Unlearnability	✗	He et al. (2022)

Unsupervised Learning. Recently, unsupervised learning has shown its great potential to learn the representation from unlabeled data. Contrastive learning, one of the popular unsupervised methods in computer vision, uses the task of instance discrimination to learn the representations. In SimCLR (He et al., 2022) which is the most common contrastive learning method, the positive and negative samples for each instance are created and the task is to discriminate the positive samples and negative samples. Some improved methods like SimSiam (Chen & He, 2021) and BYOL (Grill et al., 2020) can remove the negative samples and change the task to pushing the representations between positive samples to be similar. Many works have been proposed to explain why instance discrimination can lead to good representations. It is claimed in Wang & Isola (2020) that uniformity and alignment in feature space are the keys to a good representation.

3 Preliminary

In this section, preliminary studies are conducted to explore two types of transferability. We first introduce key notations and definitions, and then show the insufficiency of transferability. **Since the majority of unlearnable examples are generated under the supervised setting, in this work, we focus on supervised unlearnable examples and leave unsupervised unlearnable examples as one future work.** Note that the observations of supervised unlearnable examples in terms of transferability could be applicable to unsupervised unlearnable examples. For example in Appendix A, we show that unsupervised unlearnable examples also lack training-wise transferability.

3.1 Definitions

In this subsection, we first give the definition of unlearnable examples and then define the training-wise and data-wise transferability.

Unlearnable Examples. Suppose that the training dataset contains n clean examples $\mathcal{D}_c = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ with the input data $\boldsymbol{x}_i \in \mathcal{X} \subset \mathbb{R}^d$ and the associated label $y_i \in \mathcal{Y} = \{1, 2, \ldots, K\}$. We assume that the unauthorized parties will use the published training dataset to train a classifier $f_\theta : \mathcal{X} \to \mathcal{Y}$, where θ is the model parameters, and do inference on their test dataset. To protect the data from unauthorized training, instead of publishing $\mathcal{D}_c$, we want to generate an unlearnable dataset as $\mathcal{D}_u = \{(\boldsymbol{x}_i + \boldsymbol{\delta}_i, y_i)\}_{i=1}^n$ where $\boldsymbol{\delta}_i \in \Delta_{\mathcal{D}_c} \subset \mathbb{R}^d$ and $\Delta_{\mathcal{D}_c}$ is the perturbation set for $\mathcal{D}_c$. The goal of unlearnability is that if we only publish $\mathcal{D}_u$, the model f_θ trained on $\mathcal{D}_u$ performs poorly on the test dataset. Benefiting from the constraint $\|\boldsymbol{\delta}\|_p \le \epsilon$, $\mathcal{D}_u$ appears the same as $\mathcal{D}_c$ in human eyes and does not affect the normal use. The most representative EMN Huang et al. (2020) generates supervised unlearnable examples by alternating optimization on the bi-level min-min problem:

$$\min_\theta \min_{\boldsymbol{\delta}_i \in \left\{\boldsymbol{v} : \|\boldsymbol{v}\|_\infty \le \epsilon\right\}} \sum_{i=1}^n \mathcal{L}\left(f_\theta\left(\mathbf{x}_i + \boldsymbol{\delta}_i\right), y_i\right). \tag{1}$$

The outer minimization can imitate the training process, while the inner minimization can induce $\boldsymbol{\delta}_i$ to have the property of minimizing the supervised loss. Due to this property, deep models will pay more attention to the easy-to-learn $\boldsymbol{\delta}_i$ and ignore $\boldsymbol{x}_i$.

Training-wise Transferability. Supervised unlearnable examples have been designed to protect data from supervised training Huang et al. (2020). However, the unlearnable effect is almost lost when they are utilized for unsupervised training. The unauthorized parties can first get a useful feature extractor g_η from $\mathcal{D}_u$ by unsupervised training like Contrastive Learning (Chen et al., 2020a) and then fine-tunes on $\mathcal{D}_u$ or other data to get the classifier h_g. The training-wise transferability means that supervised unlearnable examples can invalidate g_η when transferred into unsupervised training.

Data-wise Transferability. When any other dataset, $\mathcal{D}_{\tilde{c}} = \{(\tilde{\boldsymbol{x}}_i, \tilde{y}_i)\}_{i=1}^{\tilde{n}}$, where $\tilde{\boldsymbol{x}}_i \in \tilde{\mathcal{X}} \subset \mathbb{R}^d$ and $\tilde{y}_i \in \tilde{\mathcal{Y}} = \{1, 2, \ldots, \tilde{K}\}$, also requires protection, it is more efficient and practical if we can transfer the perturbation that has already been generated for $\mathcal{D}_c$ onto $\mathcal{D}_{\tilde{c}}$. If the perturbation, $\Delta_{\mathcal{D}_c}$, is data-wise transferable, we can also create an unlearnable dataset $\mathcal{D}_{\tilde{u}}$ to replace $\mathcal{D}_{\tilde{c}}$ before publishing as $\mathcal{D}_{\tilde{u}} = \{(\tilde{\boldsymbol{x}}_i + \boldsymbol{\delta}_{H(i)}, \tilde{y}_i)\}_{i=1}^n$, where $\tilde{\boldsymbol{x}}_i \in \mathcal{D}_{\tilde{c}}$, $\boldsymbol{\delta}_{H(i)} \in \Delta_{\mathcal{D}_c}$, and $H(i)$ decides which perturbation in $\Delta_{\mathcal{D}_c}$ is added on $\tilde{\boldsymbol{x}}_i$ in the new dataset, $\mathcal{D}_{\tilde{c}}$. Without retraining the perturbation set, the unseen $\mathcal{D}_{\tilde{c}}$ is also protected by transferring the unlearnable perturbations from $\Delta_{\mathcal{D}_c}$.

3.2 Transferability in Existing Methods

In this subsection, we show that existing supervised unlearnable examples have almost no training-wise transferability and insufficient data-wise transferability.

3.2.1 Training-wise Transferability

In Figure 1, experiments are provided to show the ineffective protection of two supervised unlearnable examples, i.e. Error-Minimizing Noise (EMN) (Huang et al., 2020) and Synthetic Noise (SN) (Yu et al., 2021) when they are transferred into unsupervised training. We generate unlearnable examples for CIFAR-10 with EMN and SN. In supervised training, we evaluate the unlearnable dataset by the test accuracy after training with CrossEntropy loss, while in unsuperivsed training, we use SimCLR Chen et al. (2020a) to get a feature extractor and then evaluate by the accuracy of linear probing. EMN decreases the accuracy of supervised training from 93.8% to 14.7%, but only decreases the accuracy of unsupervised training by 0.2%. It means that EMN has almost no unlearnable effect after being transferred into unsupervised learning and is weak in training-wise transferability. We have similar observations for SN which also lacks training-wise transferability.

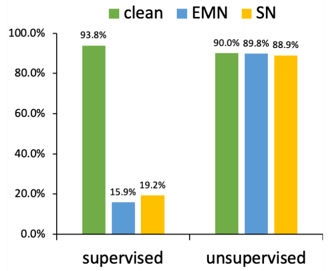
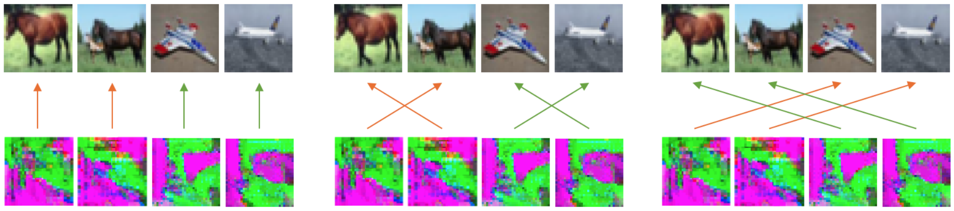


Figure 1: Accuracy on clean test data of supervised and unsupervised models on trained on clean CIFAR-10 and EMN-pertubed CIFAR-10

3.2.2 Data-wise Transferability

In this subsection, we investigate data-wise transferability for EMN and SN. As introduced in Eq. 1, EMN has been designed to induce $\boldsymbol{\delta}_i$ to have the property of minimizing the loss function in supervised training by perturbing $\boldsymbol{x}_i$. Therefore, EMN generates unlearnable examples based on



(a) original corresponding (b) intra-class swap (c) inter-class swap

Figure 2: Original one-to-one correspondence and two swapping methods.

target data. Next, we will show the data-wise transferability of EMN when the perturbations are transferred onto non-target data samples and non-target datasets. For $\boldsymbol{\delta}_i$, the target sample is $\mathbf{x}_i$, while the non-target samples are all the other samples $\mathbf{x}_j$ in $\mathcal{D}_c$, where $j \neq i$. For the whole perturbation set $\Delta_{\mathcal{D}_c}$, the non-target dataset is any other dataset $\mathcal{D}_{\tilde{c}}$. Since the perturbation $\boldsymbol{\delta}_i$ focuses on the target sample $\mathbf{x}_i$, we will first show the reduction of unlearnable effect on non-target samples. Then we will demonstrate that the perturbation set $\Delta_{\mathcal{D}_c}$ for $\mathcal{D}_c$ cannot hold the strong unlearnable effect on non-target dataset, $\mathcal{D}_{\tilde{c}}$.

On non-target samples, we change the assignment between perturbations and training samples like Fig. 2(b) within every class or Fig. 2(c) between the classes. Instead of perturbing $\boldsymbol{x}_i$ with $\boldsymbol{\delta}_i$, we add $\boldsymbol{\delta}_j$ onto $\boldsymbol{x}_i$, where $j \neq i$. Like Fig. 2(b), we construct the intra-class swapping training dataset as $\mathcal{D}_{s_{\text{intra}}} = \left\{ \left(\boldsymbol{x}_i + \boldsymbol{\delta}_{s_{\text{intra}}(i)}, y_i \right) \right\}_{i=1}^{n}$, where $s_{\text{intra}}(i)$ is the intra-class swapping function to randomly permute the correspondence between examples and perturbations within classes. For the i-th example, it uses the perturbation from another example in the same class:

$$s_{\text{intra}}(i) = j, \text{where } j \neq i, y_j = y_i,$$

Like Fig. 2(c), we construct the inter-class swapping training dataset as $\mathcal{D}_{s_{\text{inter}}} = \left\{ \left(\boldsymbol{x}_i + \boldsymbol{\delta}_{s_{\text{inter}}(i)}, y_i \right) \right\}_{i=1}^{n}$. s_{inter} permutes the corresponding between classes:

$$s_{\text{inter}}(i) = j, \text{where } j \neq i, y_j \neq y_i.$$

With $s_{\text{intra}}(i)$ and $s_{\text{inter}}(i)$, we keep the perturbation generated by EMN (i.e., $\Delta_{\mathcal{D}_c}$) and the clean data, (i.e., $\mathcal{D}_c$) unchanged, but just swap the how $\Delta_{\mathcal{D}_c}$ corresponds to $\mathcal{D}_c$. As shown in Table 2, the unlearnable effect decreases significantly under intra-class swapping and inter-class swapping.

EMN unlearnable samples are also limited in data-wise transferability on non-target datasets. We first generate the perturbations $\Delta_{\mathcal{D}_c}$ on CIFAR-10 and then choose another non-target dataset SVHN-small, which is downsampled from SVHN(Netzer et al., 2011), as non-target dataset $\mathcal{D}_{\tilde{c}}$ to protect. CIFAR-10 has ten classes, where every class has 5,000 training images. SVHN also has ten classes, but some classes have more than 5,000 training images. Thus we sample 5,000 training images in every class from SVHN and construct SVHN-small. In Table 3, we first generate EMN on SVHN-small to protect target dataset, i.e. SVHN-small. It protects the data from unauthorized training and reduces the test accuracy to only 11.64%. But when we use EMN generated on CIFAR-10 to protect SVHN-small, SVHN-small becomes a non-target dataset. More information is learned from SVHN-small and the test accuracy is 27.59%, which means the non-target dataset gets less protection.

Table 2: Comparison of supervised unlearnable examples on target and non-target samples.

Corresponding	CIFAR10		CIFAR100	
	EMN	SN	EMN	SN
Original	15.88	14.07	6.59	2.13
Intra-class swap	30.74	13.59	21.63	2.44
Inter-class swap	30.15	13.15	35.50	2.73

Table 3: Comparison of supervised unlearnable examples on target and non-target datasets.

Generated on	Tested on SVHN-small	
	EMN	SN
SVHN-small	11.64	8.46
CIFAR-10	27.59	9.58

We conducted similar experiments for SN Yu et al. (2021). The results for non-target samples and non-target datasets are illustrated in Table 2 and Table 3, respectively. We observe that SN is data-wise transferable. According to Yu et al. (2021), SN can create linear separability that the label-related imperceptible perturbation is linearly separable between different classes, as a shortcut for the optimization objective. In other words, the linear separability is only between $\boldsymbol{\delta}_i$ and y_i, and not related to $\boldsymbol{x}_i$. Therefore, the SN perturbation which is totally based on linear separability can lead to the unlearnable effect and its unlearnability has good data-wise transferability. However, SN is generated by sampling from a manually designed distribution and the sampling process is incompatible with an optimization process. Thus, it is hard to directly incorporate SN into other optimization objectives to enjoy linear separability for data-wise transferability.

3.3 Discussions

As shown by the above preliminary studies, both EMN and SN lack training-wise transferability, and EMN has insufficient data-wise transferability. Meanwhile, we found that the linearly separable SN perturbation that is independent on the data has good data-wise transferability. However, the manually designed linear separability of SN is unable to be leveraged by other optimization objectives. Thus in the next section, we propose Classwise Separability Discriminant and a new framework to generate unlearnable examples with training-wise and data-wise transferability.

4 Transferable Unlearnability from Classwise Separability Discriminant

In order to improve the two types of transferability of unlearnable examples, we propose the optimizable Classwise Separability Discriminant (CSD) to quantify linear separability. Furthermore, we propose Transferable Unlearnbale Examples (TUE) that have superior training-wise and data-wise transferability. It not only generalizes the protection scenario from supervised to unsupervised training but also maintains the unlearnability when transferred to non-target datasets.

4.1 Classwise Separability Discriminant

The linear separability lies in two factors, i.e. intra-class distance and inter-class distance. Intuitively, better linear separability usually has smaller intra-class distance and larger inter-class distance. Because smaller intra-class distance means that the perturbations from the same class can concentrate on a small area in the input space, while larger inter-class distance means that the perturbations of different classes are far away from each other. When the overlapping area between different classes is reduced by smaller intra-class distance and larger inter-class distance, the perturbations can become features that are easily separated by even a linear classifier. For measuring the intra-class and inter-class distance, we first define the centroid of the perturbations of every class: $\boldsymbol{c}_k = \frac{1}{|\{\boldsymbol{\delta}_i:y_i=k\}|}\sum_{\{\boldsymbol{\delta}_i:y_i=k\}}\boldsymbol{\delta}_i$, where $\{\boldsymbol{\delta}_i : y_i = k\}$ is the perturbations that belong to class k. Then we have the intra-class distance as:

$$\boldsymbol{\sigma}_k = \frac{1}{|\{\boldsymbol{\delta}_i : y_i = k\}|} \sum_{\{\boldsymbol{\delta}_i:y_i=k\}} d(\boldsymbol{\delta}_i, \boldsymbol{c}_k), \tag{2}$$

where $d(\cdot,\cdot)$ is the Euclidean distance. $\boldsymbol{\sigma}_k$ measures the average distance between the perturbation $\boldsymbol{\delta}_i$ whose label is k to the centroid $\boldsymbol{c}_k$. The inter-class distance is defined by the Euclidean distance between two centroids:

$$d_{i,j} = d\left(\boldsymbol{c}_i, \boldsymbol{c}_j\right) \tag{3}$$

To enhance the linear separability with an optimizable objective function, we propose Classwise Separability Discriminant (CSD):

$$\mathcal{L}_{\mathrm{S}}(\{\boldsymbol{\delta}_i, y_i\}_{i=1}^{n}) = \frac{1}{M}\sum_{i=1}^{M}\frac{1}{M-1}\sum_{j\neq i}^{M-1}\left(\frac{\boldsymbol{\sigma}_i + \boldsymbol{\sigma}_j}{d_{i,j}}\right), \tag{4}$$

where M is the number of classes, and y_i represents the ground truth label for sample $\boldsymbol{x}_i \in \mathcal{D}_c$. It is worth mentioning that $\mathcal{L}_{\mathrm{S}}(\{\boldsymbol{\delta}_i, y_i\}_{i=1}^{n})$ is not related to $\boldsymbol{x}_i$, which means that $\boldsymbol{\delta}_i$ is generated independently on $\boldsymbol{x}_i$. $\boldsymbol{\sigma}_i$ and $\boldsymbol{\sigma}_j$ measure the intra-class distance of class i and class j, while $d_{i,j}$ measures the inter-class distance between class i and class j. The ratio between intra-class distance and inter-class distance, $\frac{\boldsymbol{\sigma}_i+\boldsymbol{\sigma}_j}{d_{i,j}}$, can show the overlapping between two classes. Better clustering distribution has smaller intra-class distance and larger inter-class distance, which lead to a lower ratio and smaller overlapping area between two classes. Therefore, lower Classwise Separability Discriminant, which indicates more compact intra-class distance and better farther inter-class distance, has better clustering effect and linear separability.

4.2 Transferable Unlearnable Examples

As discussed in Section 3.2.2, linear separability plays an essential role in supervised unlearnability and data-wise transferability. In order to enhance the data-wise and training-wise transferability of unlearnable examples simultaneously, we aim to incorporate linear separability into unsupervised unlearnable examples to propose Transferable Unlearnable Examples (TUE) framework. In particular, TUE uses contrastive learning as the unsupervised backbone and embeds linear separability via Classwise Separability Discriminant into unsupervised unlearnability. Most unsupervised contrastive learning algorithms, such as SimCLR (Chen et al., 2020a), MoCo (Chen et al., 2020b) and SimSiam (Chen & He, 2021), enforce the similarity between two augmentations of the same samples such that the model can achieve representation learning. TUE generates the perturbation that not only promotes this similarity but also has linear separability by the bi-level optimization problem as follows:

$$\min_{\theta} \min_{\{\boldsymbol{\delta}_i:\|\boldsymbol{\delta}_i\|_\infty\le\epsilon\}} \sum_{i=1}^{n} \mathcal{L}_{\mathrm{CL}}\Big(f\big(\theta, T_1(\boldsymbol{x}_i+\boldsymbol{\delta}_i)\big), f\big(\theta, T_2(\boldsymbol{x}_i+\boldsymbol{\delta}_i)\big)\Big) + \lambda\mathcal{L}_{\mathrm{S}}(\{\boldsymbol{\delta}_i, y_i\}_{i=1}^{n}), \quad (5)$$

where $\mathcal{L}_{\mathrm{CL}}$ is the loss of contrastive learning, and λ is the weight to balance between two loss terms. T_1 and T_2 are augmentations on input data. The first term in Eq. 5 can ensure that the generated perturbation has the property of promoting the similarity between two views of augmented sample to reduce $\mathcal{L}_{\mathrm{CL}}$. In this way, the perturbation provides easy-to-learn shortcut information for unsupervised learning such that the model will only learn to extract the perturbation instead of the intrinsic semantic information in the data. In other words, the first term is to ensure the unlearnability for unsupervised training. The second term can enhance the linear separability of the perturbations. Furthermore, we propose to solve this bi-level optimization problem in Eq. 5 alternately:

$$\begin{cases} \mathrm{S1}: \theta^{(t)} = \arg\min\limits_{\theta} \sum\limits_{\boldsymbol{x}_i\in\mathcal{D}_c} \mathcal{L}_{\mathrm{CL}}\left(f\big(\theta, T_1(\boldsymbol{x}_i+\boldsymbol{\delta}_i^{(t-1)})\big), f\big(\theta, T_2(\boldsymbol{x}_i+\boldsymbol{\delta}_i^{(t-1)})\big)\right) \\ \mathrm{S2}: \boldsymbol{\delta}_i^{(t)} = \arg\min\limits_{\{\boldsymbol{\delta}_i:\|\boldsymbol{\delta}_i\|_\infty\le\epsilon\}} \mathcal{L}_{\mathrm{CL}}\left(f\big(\theta^{(t)}, T_1(\boldsymbol{x}_i+\boldsymbol{\delta}_i)\big), f\big(\theta^{(t)}, T_2(\boldsymbol{x}_i+\boldsymbol{\delta}_i)\big)\right) + \lambda\mathcal{L}_{\mathrm{S}}(\{\boldsymbol{\delta}_i, y_i\}_{i=1}^{n}). \end{cases} \quad (6)$$

In the first step (S1), we update the model parameter θ to minimize the unsupervised loss $\mathcal{L}_{\mathrm{CL}}$, while in the second step (S2) we optimize the perturbation $\{\boldsymbol{\delta}_i\}$ to jointly reduce the unsupervised loss and force the linear separability among different classes. By the bi-level optimization on unsupervised loss and Classwise Separability Discriminant, we can generate unlearnable examples with both data-wise and training-wise transferability.

Interpolation for data-wise transferability. Once we have finished the generation process on the training dataset, $\mathcal{D}_c$, we can use this perturbation without change on another dataset, $\mathcal{D}_{\tilde{c}}$, to make it unlearnable as well. Nevertheless, it is possible that the number of classes in $\mathcal{D}_c$ or the number of samples in one class in $\mathcal{D}_c$ is less than $\mathcal{D}_{\tilde{c}}$. The generated perturbation may not cover every sample in $\mathcal{D}_{\tilde{c}}$. To solve this problem, we use interpolation to create more perturbations. Interpolation can make use of current perturbation in $\Delta_{\mathcal{D}_c}$ to enlarge its size and transfer to a larger dataset. If more classes are desired, we can interpolate between two current classes in $\mathcal{D}_c$ to create new classes:

$$\boldsymbol{x}_k^* = \alpha\boldsymbol{x}_i + (1-\alpha)\boldsymbol{x}_j, \text{where } y_i \neq y_j. \quad (7)$$

If more samples in one class are required, we can interpolate within current classes in $\mathcal{D}_c$ to create new samples:

$$\boldsymbol{x}_k^* = \alpha\boldsymbol{x}_i + (1-\alpha)\boldsymbol{x}_j, \text{where } y_i = y_j. \quad (8)$$

By varying α, more than one sample can be created from the interpolation between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. Empirical results in Section 5.3 show that this interpolation strategy works very well.

5 Experiment

In this section, we validate the data-wise and training-wise transferability of the proposed TUE. We first introduce the experimental setups in Section 5.1. In Section 5.2, we present the experimental results on the training-wise transferability. In Section 5.3 we demonstrate that TUE has improved data-wise transferability. In Section 5.4 we illustrate the enhanced linear separability in input space by visualization. Finally, in Section 5.5 we show that the two types of transferability make TUE have both classwise and samplewise characteristics as expected by case study.

5.1 Experimental Setups

Datasets. The datasets include CIFAR-10 and CIFAR-100 (Krizhevsky et al., 2009), which contain 50,000 training images and 10,000 test images, and SVHN (Netzer et al., 2011), which contains 73,257 training images of ten classes and 26,032 test images. We randomly sample from SVHN to construct SVHN-small where the number of training images in every class is no more than 5000.

Generation process and Evaluation settings. We compare our method with representative unlearnable methods in both supervised and unsupervised training. All the perturbations are generated by

PGD (Madry et al., 2018) on ResNet-18 and constrained by $\|\delta_i\|_\infty \leq \epsilon$ where $\epsilon = 8/255$. We use CrossEntropy as the objective function in supervised training. and linear probing after contrastive pre-training in unsupervised training. The supervised model is trained for 200 epochs. The unsupervised model is pre-trained for 1000 epochs and then fine-tuned for 100 epochs. More hyperparamaters for generation process and evaluation can be found in Appendix B.

Baselines. We use three representative unlearnable examples as baselines, which are Error-Minimizing Noise (EMN) (Huang et al., 2020), Unlearnable Contrastive Learning (UCL) (He et al., 2022) and Synthetic Noise (SN) (He et al., 2022).

Backbones. We tested TUE on three different unsupervised backbones, SimCLR (Chen et al., 2020a), MoCo (Chen et al., 2020b) and SimSiam (Chen & He, 2021). The backbones are also used in UCL.

Table 4: Performance of unlearnable effects on different algorithms and datasets in supervised and unsupervised training. This table reports the accuracy (%) of supervised training and the accuracy (%) of linear probing after contrastive pre-training (SimCLR, MoCo, and SimSiam).

Dataset	CIFAR-10		CIFAR-100	
SimCLR	Supervised	Unsupervised	Supervised	Unsupervised
Clean Data	93.79	90.04	74.49	63.68
EMN	14.74	89.79	5.23	62.00
SN	19.23	88.93	2.13	62.31
UCL	92.86	**47.78**	72.17	**16.68**
TUE	**10.67**	52.38	**0.76**	19.51
MoCo	Supervised	Unsupervised	Supervised	Unsupervised
Clean Data	93.79	89.90	74.49	63.03
EMN	14.74	89.18	5.23	60.62
SN	19.23	89.32	2.13	61.81
UCL	92.62	**44.24**	71.59	**18.74**
TUE	**10.06**	63.38	**1.09**	23.6
SimSiam	Supervised	Unsupervised	Supervised	Unsupervised
Clean Data	93.79	90.59	74.49	64.69
EMN	14.74	91.43	5.23	65.96
SN	19.23	91.54	2.13	66.83
UCL	93.50	**30.43**	71.84	**4.64**
TUE	**10.03**	35.57	**1.21**	6.17

5.2 TRAINING-WISE TRANSFERABLE UNLEARNABILITY

As mentioned above, existing unlearnable methods focus on one training setting, either supervised or unsupervised training. In this work, we use the proposed TUE to make the dataset unlearnable in both two training settings and thus protect the data from unauthorized training in a comprehensive way. In Table 4, we reported the comparison on CIFAR-10 and CIFAR-100 with supervised training by CrossEntropy Loss and three unsupervised training method, SimCLR (Chen et al., 2020a), MoCo (Chen et al., 2020b) and SimSiam (Chen & He, 2021). The unlearnable examples are evaluated by two training settings, supervised and unsupervised training. We use TUE to embed the linear separability into SimCLR, MoCo and SimSiam, respectively, and get three protected datasets. When testing with unsupervised training, we use the corresponding unsupervised algorithm that is used to generate TUE.

Table 4 shows the model accuracy on clean test data after being trained on unlearnable data. First, we observe that previous unlearnability methods can only work under one training setting, and have almost no protection under the other training setting. Specifically, EMN and SN reduces the test accuracy of supervised training to less than 20% on CIFAR-10 and less than 6% on CIFAR-100 but cannot transfer well in all of the three unsupervised trainings. UCL reduces the accuracy of unsupervised training to less than 50% on CIFAR-10 and less than 20% on CIFAR-100 but has almost no protection from supervised training. Second, TUE can hold the unlearnability in both supervised and unsupervised training settings. It reduces the supervised accuracy to around 10% on CIFAR-10 and 1% on CIFAR-100 and reduces the unsupervised accuracy to a comparable level to UCL. In particular, it can even perform better than EMN in supervised training. The test accuracy of unauthorized classifiers trained on TUE is around 4% lower than EMN on both CIFAR-10 and

CIFAR-100. Because the enhanced linear separability causes that supervised model focuses more on TUE perturbation than EMN perturbation. In summary, from Table 4, we can see that TUE has good training-wise transferability, which can maintain the unlearnable effect when being transferred from supervised training to unsupervised training and it is the only one that maintains the unlearnable effects under two training settings.

5.3 Data-wise transferable unlearnability

After demonstrating the training-wise transferability of TUE, in this subsection, we show that TUE also has better data-wise transferability, i.e., the perturbations generated with TUE on one dataset can also protect any other datasets from unauthorized supervised training. Following the experimental settings in the preliminary studies of Section 3.2.2, we test the data-wise transferability first on non-target data samples, and then on non-target datasets. The results are reported in Table 5 and Table 6, respectively.

Table 5: Test accuracy (%) with different correpondings between train data and perturbations.

Dataset	Methods	Original	Intra	Inter
CIFAR-10	EMN	15.88	30.74	33.91
	SN	14.07	13.59	13.15
	TUE (SimCLR)	10.67	10.16	10.90
	TUE (MoCo)	10.06	12.04	8.57
	TUE (SimSiam)	10.03	10.25	10.47
CIFAR-100	EMN	6.59	21.63	35.50
	SN	2.13	2.44	2.73
	TUE (SimCLR)	0.76	1.11	1.13
	TUE (MoCo)	1.09	1.25	3.63
	TUE (SimSiam)	1.21	1.28	1.08

Table 6: Test accuracy (%) of transferring the perturbation generated on CIFAR-10 to different datasets.

	SVHN-small	CIFAR-100	SVHN
EMN	27.59	21.80	24.72
SN	9.58	9.35	7.77
TUE (SimCLR)	9.77	10.53	11.72
TUE (MoCo)	11.29	8.32	13.95
TUE (SimSiam)	10.28	5.10	12.93

First, for non-target data samples, we compare the performance under the original correspondence, intra-class swapping and inter-class swapping between samples and perturbations in the same dataset. In Table 5, TUE can maintain similar test accuracy under different correspondences on CIFAR-10 and CIFAR-100. For example, on CIFAR-100, the test accuracy of EMN in intra-class and inter-class swapping is around, respectively, 15% and 29% higher than that in the original correspondence, but for TUE, the difference in the test accuracy under different correspondences is less than 3%. Although we can also notice this data-wise transferability in SN, it is not an optimizable method and is unable to be used for improving the training-wise transferability.

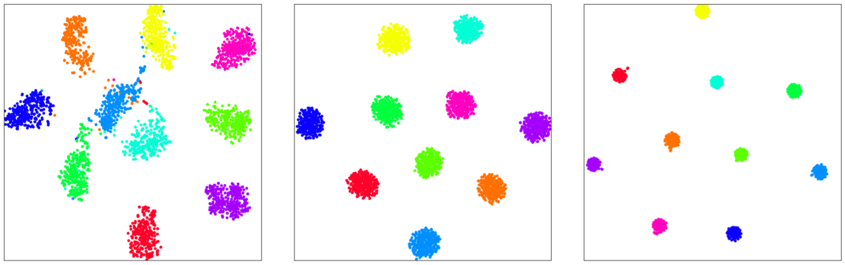
Second, to verify that the proposed TUE can maintain unlearnability when transferred to different datasets, we generate TUE on CIFAR-10 and test the performance of transferring onto three datasets, SVHN-small, CIFAR-100 and SVHN. TUE on CIFAR-10 can be directly transferred to SVHN-small without interpolation, while CIFAR-100 requires interpolation for more classes and SVHN requires interpolation for more samples. The baseline methods are also interpolated for comparison. In Table 6, we can observe that on all the datasets, TUE can maintain the unlearnable effect after being transferred onto all the other datasets. The enhanced linear separability of TUE ensures that the test accuracy is low. For example, EMN has poorer protection after transferred onto SVHN-small with the accuracy of 27.59%, but TUE can maintain the unlearnability on non-target datasets for all the backbones at around 10% accuracy.

From Table 5 and Table 6, we can draw the conclusion that the proposed TUE has better data-wise transferability than optimization-based unlearnable methods like EMN, and our strategy to transfer TUE to protect other datasets can be very efficient.

5.4 Linear Separability of Transferable Unlearnable Examples

In this subsection, we aim to better understand how linear separability influences unlearnable examples by visualization. To better visualize linear separability of the perturbations, we use t-SNE to show the perturbation in the input space. Although t-SNE cannot accurately describe the high-dimension space, it is useful to observe the separability of the perturbations in the input space. We compare the perturbations in the input space in the following three groups.

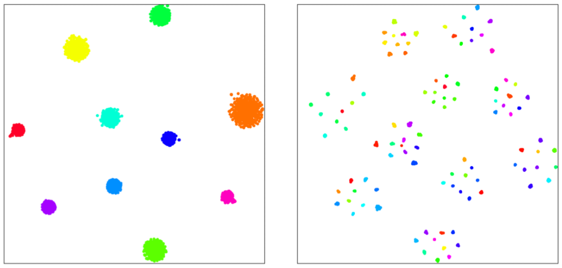
Comparison of Linear separability in supervised unlearnable examples. In Fig. 3, we show the perturbations of EMN, SN and TUE to understand why linear separability is more effective than EMN when used in supervised classification. In Fig. 3, the perturbations of different classes in TUE and SN are clearly separated and have no overlapping. In contrast, EMN has some data points mixed



(a) EMN (b) SN (c) TUE

Figure 3: t-SNE visualization of linear separability of supervised unlearnable examples.

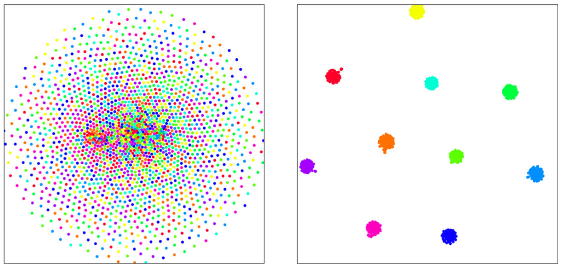
with other classes which are confusing when used in classification. Meanwhile the clusters in EMN do not contract as well as TUE and SN. Since better linear separability can provide useful information about classification, the supervised model will focus on TUE and SN more than EMN and ignore the semantic features in clean images.



(a) Interpolation: SVHN (b) Interpolation: CIFAR-100

Figure 4: t-SNE visualization of linear separability of two examples of interpolation.

Linear separability in Interpolation. In Fig. 4, we show that the interpolation on TUE from CIFAR-10 to SVHN and CIFAR-100 can also hold linear separability which makes it data-wise transferable. We can observe that in the interpolation for SVHN, although the size of each class is different, they can still hold clear linear separability and unlearnbility. In SVHN, the interpolation within classes creates more samples for one class. Since the numbers of examples in different classes in SVHN are different, it can be observed that some clusters are much larger than others. For CIFAR-100, more new classes are created and they still keep good linear separability. So the supervised model trained on interpolation-based unlearnable SVHN and CIFAR-100 will learn nothing but the perturbations and perform poorly on clean test data which has no perturbations.



(a) UCL (b) TUE

Figure 5: t-SNE visualization of linear separability of supervised and unsupervised unlearnability.

Comparison of Linear separability between supervised and unsupervised unlearnability. In Fig. 5, we show the perturbations of UCL and TUE in input space to understand why UCL cannot help with unlearnable effects in supervised training. TUE is linear separable, which can be used to protect the data from both supervised and unsupervised training. But for UCL, all the perturbations are mixed, they cannot provide the class information for supervised training. So for a supervised model, UCL is not a helpful feature and will not vanquish the semantic feature.

5.5 Case Study

In this subsection, by showing the patterns of the perturbations generated by different methods, we can deepen our understanding on linear separability in TUE and observe the classwise and samplewise characteristics, which reflect the basic idea behind training-wise transferability.

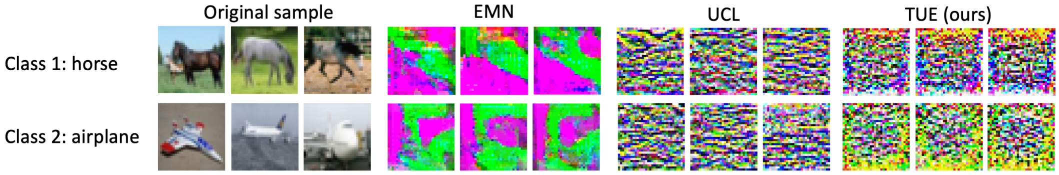


Figure 6: Visualization of the unlearnable examples of EMN, UCL and TUE(ours). We rescale the perturbation from [-8/255, 8/255] to [0,1] for better visualization.

We illustrate the samples and unlearnable perturbations from two classes of CIFAR-10, i.e., horse and airplane, in Figure 6. First, we find that EMN and TUE both have classwise noise patterns. Every noisy sample has similar features in one class, while every class has a different feature pattern from other classes. This is a new perspective to show that TUE has linear separability and perturbations from the same class are clustered in input space. Second, in UCL, the perturbations can be more diverse and related to each sample, which indicates that unsupervised unlearnability requires more complex feature patterns. We cannot see the classwise patterns like EMN. Because UCL uses no label information and has no concept of classes. Third, we notice that TUE has both classwise and samplewise characteristics as expected. The classwise feature can be the easy-to-learn feature for the supervised model, while the samplewise feature can be effective in reducing unsupervised loss, which makes the unlearnability transferable from supervised training to unsupervised training. In summary, from the case study, we can observe that the co-existing classwise and samplewise features provide supervised and unsupervised protection at the same time. The linear separability makes the supervised unlearnability data-wise transferable.

6 Conclusion

We reveal the limitation of training-wise and data-wise transferability in existing unlearnable examples and propose Classwise Separability Discriminant to enhance transferability. The proposed Transferable Unlearnable Examples (TUE) can be transferred from supervised to unsupervised training with the unlearnable effect maintained and can protect data from unauthorized usage in a comprehensive way. Meanwhile, the unlearnability of TUE is transferable across datasets. It can generate unlearnbility once for all, which makes unlearnable examples more practical and efficient. TUE greatly pushes the boundary of existing unlearnable methods that can only work on the target data and target training setting. We improve the data-wise and training-wise transferability of unlearnable examples and provide more flexible, practical, and comprehensive protections from unauthorized data usage.

References

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pp. 1247–1250, 2008.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020a.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15750–15758, 2021.

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020b.

Liam H Fowl, Micah Goldblum, Ping-yeh Chiang, Jonas Geiping, Wojciech Czaja, and Tom Goldstein. Adversarial examples make strong poisons. In *Advances in Neural Information Processing Systems*, 2021.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020.

Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *European conference on computer vision*, pp. 87–102. Springer, 2016.

Hao He, Kaiwen Zha, and Dina Katabi. Indiscriminate poisoning attacks on unsupervised contrastive learning. *arXiv preprint arXiv:2202.11202*, 2022.

Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In *Workshop on faces in'Real-Life'Images: detection, alignment, and recognition*, 2008.

Hanxun Huang, Xingjun Ma, Sarah Monazam Erfani, James Bailey, and Yisen Wang. Unlearnable examples: Making personal data unexploitable. In *International Conference on Learning Representations*, 2020.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

Luis Muñoz-González, Battista Biggio, Ambra Demontis, Andrea Paudice, Vasin Wongrassamee, Emil C Lupu, and Fabio Roli. Towards poisoning of deep learning algorithms with back-gradient optimization. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, pp. 27–38, 2017.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

Shawn Shan, Emily Wenger, Jiayun Zhang, Huiying Li, Haitao Zheng, and Ben Y Zhao. Fawkes: Protecting privacy against unauthorized deep learning models. In *29th USENIX Security Symposium (USENIX Security 20)*, pp. 1589–1604, 2020.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pp. 9929–9939. PMLR, 2020.

Da Yu, Huishuai Zhang, Wei Chen, Jian Yin, and Tie-Yan Liu. Indiscriminate poisoning attacks are shortcuts. *arXiv preprint arXiv:2111.00898*, 2021.

A Existing unlearnable methods cannot protect unlabeled dataset in a training-wise trasnferable way

Previous unlearnable methods can only protect data from one unauthorized training setting. According to whether the dataset is annotated, unlearnable examples can be classified in two settings, supervised and unsupervised unlearnable examples, as mentioned in Section 2. The supervised unlearnable methods, like EMN, can create unlearability based on the label information and protect the data from supervised training, but these methods cannot prevent the unauthorized parties from training the data with an unsupervised algorithm. Similarly, the unsupervised unlearnable examples produced by Unleanrable Contrastive Learning (UCL) He et al. (2022) can destroy the unsupervised training but they are still learnable in the supervised training. In Fig. 7, EMN decreases the accuracy of supervised training from 93.8% to 14.7%, but only decreases the accuracy of unsupervised training by 0.2%. Similarly, SN decreases supervised training to 14.1%, but only decreases unsupervised training by 1.1%. Supervised unlearnable examples have almost no training-wise transferability to unsupervised training. UCL, which produces unsupervised unlearnable examples, can protect data from unsupervised training, but cannot be transferred well to supervised training. Under unsupervised training, UCL can decrease the test accuracy from 90.0% to 47.8%, but under supervised training, it can only decrease the accuracy by 0.9%.

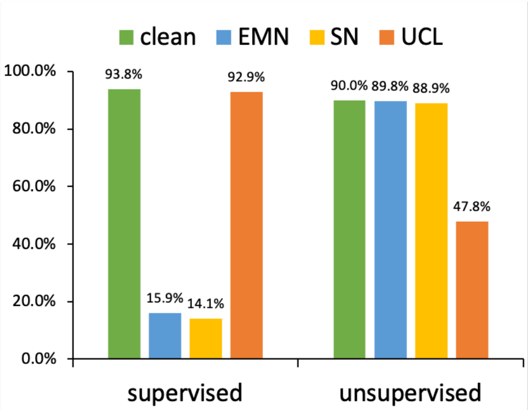


Figure 7: Accuracy on test data of classifiers produced by supervised training and unsupervised training on the unlearnable examples from CIFAR-10

Table 7: Hyperparameters of evaluation stage

Hyperparameter	Supervised	Unsupervised pretraining			Linear probing		
		SimCLR	MoCo	SimSiam	SimCLR	MoCo	SimSiam
Epoch	200	1000	1000	1000	100	100	100
Optimizer	SGD	SGD	Adam	SGD	Adam	SGD	SGD
Learning Rate	0.1	0.06	0.3	0.06	0.001	30	30
LR Scheduler	CosineAnnealingLR (CA)	-	CA	CA	-	CA	CA
Encoder Momentum	-	-	0.99	-	-	-	-
Loss Function	CrossEntropy (CE)	InfoNCE	InfoNCE	Similarity between positive samples	CE	CE	CE

B Details of experimental settings

B.1 Settings of generation process

All the baselines and our proposed TUE are generated in the way of PGD Madry et al. (2018), except for SN, which is sampled from a manually designed distribution following the algorithm in Yu et al. (2021). For EMN, UCL, and TUE, the generation process is an alternate optimization between model parameters and perturbations. For example, TUE is optimized in the way of Eq. 6. In EMN, after every epoch of optimization on model parameters, the whole set perturbations are optimized for one epoch by PGD-20. In TUE, different backbones are set to different schedule. For TUE (SimCLR), the model parameters are trained for 40 epochs, and after every 1/5 epoch of optimization on model parameters, the whole set perturbations are optimized for one epoch by PGD-20. For TUE (MoCo), the model parameters are trained for 200 epochs, and after every epoch of optimization on model parameters, the whole set perturbations are optimized for one epoch by PGD-5. For TUE (SimSiam), the model parameters are trained for 50 epochs, and after every 1/4 epoch of optimization on model parameters, the whole set perturbations are optimized for one epoch by PGD-20. UCL with SimCLR and SimSiam have the same settings as TUE. UCL with MoCo uses PGD-10 and the other settings in the schedule is the same as TUE (MoCo). Finally, all the perturbations are constrained by l_∞ norm, i.e. $\|\delta_i\|_\infty \leq \epsilon$ where $\epsilon = 8/255$.

B.2 Settings of evaluation stage

To evaluate the unlearnable effect, we use CrossEntropy as the objective function in the unauthorized supervised training. We use linear probing after contrastive pre-training to evaluate the unlearnable effect in unsupervised training. The supervised model is trained for 200 epochs. The unsupervised model is pre-trained for 1000 epochs by unsupervised contrastive learning and then fine-tuned for 100 epochs by linear probing. The details for the hyperparameters are listed in Table 7.