UNIVERSAL UNLEARNABLE EXAMPLES: CLUSTER-WISE PERTURBATIONS WITHOUT LABEL-CONSISTENCY

Anonymous authors
Paper under double-blind review

ABSTRACT

There is a growing interest in employing unlearnable examples against privacy leaks on the Internet, which prevents the unauthorized models from being properly trained by adding invisible image noise. However, existing attack methods rely on an ideal assumption called label-consistency. In this work, we clarify a more practical scenario called *label-inconsistency* that allows hackers and protectors to hold different labels for the same image. Inspired by disrupting the *uniformity* and *discrepancy*, we present a novel method called *UniversalCP* on label-inconsistency scenario, which generates the universal unlearnable examples by cluster-wise perturbation. Furthermore, we investigate a new strategy for selecting the CLIP as the surrogate model, since vision-and-language pre-training models are trained on large-scale data and more semantic supervised information. We also verify the effectiveness of the proposed methods and the strategy for selecting surrogate models under a variety of experimental settings including black-box backbones, datasets and even commercial platforms Microsoft `Azure` and Baidu `PaddlePaddle`.

1 INTRODUCTION

It has been observed that the unconscious and unauthorized collection of users' images from Internet for training commercial models raises privacy problems (Hill, 2020). E.g., a private company called Clearview AI was disclosed to unconsciously crawl more than three billion users' images from Facebook, YouTube and other websites to construct a commercial API (Zhang et al., 2020). To address this concern, *unlearnable examples* are proposed to make training examples unusable for Deep Neural Networks (DNNs) (Huang et al., 2021). In other literature, they are also known as availability attacks (Biggio & Roli, 2018; Yu et al., 2022) or indiscriminate poisoning attacks (He et al., 2022), which imperceptibly perturb the training images to injure the DNNs' performance during inference phase.

However, these methods are generally based on the ideal assumption: *the labels used for the same image to train the surrogate model and the victim model*[1] *are identical*. The protectors transform a clean dataset $\mathcal{D}_c = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}$ with images $\boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^d$ and labels $\boldsymbol{y} \in \mathcal{Y}$ into an unlearnable dataset $\mathcal{D}_u = \{(\boldsymbol{x}'_i, \boldsymbol{y}'_i)\}$ with unlearnable images $\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{\delta}$ and $\boldsymbol{x} \in \mathcal{D}_c$ and imperceptible perturbation $\boldsymbol{\delta} \in \Delta \subset \mathbb{R}^d$ and labels $\boldsymbol{y}' \in \Omega$. The flaw in this ideal assumption is the belief that $\Omega = \mathcal{Y}$, yet this is almost impossible in practice, since *the protector only releases unlabeled images to the hacker via the web*. For instance, the label $\boldsymbol{y}_i$ of an image $\boldsymbol{x}_i$ in $\mathcal{Y}$ is a "domestic cat", but the label $\boldsymbol{y}'$ in Ω could be a "cat" (hyperonymy relation) or an "American shorthair" (hyponymy relation), or even a "black-footed cat" (similar relation). In other words, for the same images, the labels owned by the protector and the labels owned by the hacker are inconsistent. We define the former ideal assumption as the *label-consistency* scenario, and the more realistic latter as the *label-inconsistency* scenario. We found the current methods suffer from a significant reduction in effectiveness in the latter (Figure 1), because the dependency for these methods on specific labels derives invalidation in label-inconsistency scenarios.

Therefore, we are faced with the need to find ways that can decouple the poisoning attacks from the label dependencies to fit the label-inconsistency scenario. To this end, we analyzed the SOTA

[1]The surrogate model refers to the model that generates the unlearnable examples. The victim model refers to the model that is trained on unlearnable examples, i.e., poisoned by unlearnable examples. The protector uses the surrogate models to poison the victim models belonging to the hacker.

attacks in label-consistency scenario (Error-minimizing Noise (Huang et al., 2021) and Adversarial Poisoning (Fowl et al., 2021)) and sought to find the common core mechanisms behind them. By observing the visual analysis, we found that the reason why these powerful poisoning attacks work is that they disrupt the uniformity and discrepancy of the data distribution in feature representations (Figure 2). *Uniformity* refers to the property that the manifold of unlearnable examples in the feature space does not deviate from clean examples, and *discrepancy* refers to the property that examples belonging to the same subject are richly diverse in the feature space. Inspired by this, we propose the Universal unlearnable examples with Cluster-wise Perturbation (*UniversalCP*) to solve the label-inconsistency issue. This allows us to achieve disrupting uniformity and discrepancy simultaneously without knowing the labelled information.

Another issue beyond label-inconsistency is how to choose an appropriate surrogate model. It is supposed to be a classified DNN that contains as many classes as possible in order to facilitate the recognition and protection of billions of images on the Internet, such as the ImageNet (Russakovsky et al., 2015) model (Huang et al., 2021). Yet, since we have removed the limitation on label dependency, we can extend the choice of surrogate models to numerous non-classified models. Among them, vision-and-language pre-training models (VLPMs) (Li et al., 2019; Radford et al., 2021; Li et al., 2021) enjoy strong representation capabilities (more precisely, image encoders of VLPMs) because they are trained on a large amount of training images and more semantic supervision (using a textual description to align with the image rather than a one-hot label). In this paper, we propose to employ CLIP[2] (Radford et al., 2021) as surrogate models to generate unlearnable examples.

To solidly demonstrate the effectiveness of our methods in practical applications, we designed more black-box settings, including several black-box backbones and black-box datasets. More importantly, we compared our methods with baselines on two commercial machine learning platforms: Microsoft `Azure`[3] and Baidu `PaddlePaddle`[4]. To our knowledge, this is the first to conduct experimental evaluations on commercial APIs in this line of work. Our contributions are summarized as follows:

- We clarify a more practical scenario called *label-inconsistency* in using imperceptible perturbations to protect user privacy from unconscious and unauthorized collection, in which the existing baseline methods are rendered inefficient.
- We analyze the existing attacks and attribute the common core mechanism to disrupting the *uniformity* and *discrepancy*. Then we propose a novel method called *UniversalCP*, which allows us to generate the universal unlearnable examples by cluster-wise perturbation without knowing the labelled information.
- We present a novel strategy for choosing surrogate models. Since VLPMs are trained on large-scale data and more semantic supervised information, we explore the employment of CLIP as the surrogate model.
- We empirically verify the effectiveness of the proposed methods by designing a variety of black-box settings including backbones and datasets. We also demonstrate the practical performances in real-world scenarios via the commercial platforms.

2 Related Work

Poisoning Attacks. Poisoning attacks refer to perturbing some or the entire training dataset so that the model performs poorly on the entire validation dataset or some certain samples (Biggio et al., 2012; Biggio & Roli, 2018). Considering the difference in the perturbing objects, the poisoning attacks can be divided into clean-label attacks (perturbing images) and dirty-label attacks (perturbing labels) (Yuan & Wu, 2021). Among the latter, the attacks that degrade the performance of the model over the entire validation dataset by adding imperceptible image noise are known as indiscriminate poisoning attacks or availability attacks or unlearnable examples (Huang et al., 2021; Fowl et al., 2021; Yu et al., 2022; He et al., 2022). In this paper, we focus on these attacks since they maintain image perception and also prevent hackers from using these perturbed data to train their own models.

[2]More precisely, we just employ the image encoder of CLIP in this work. For brevity, we use this description in the rest of this paper.

[3]`https://portal.azure.com/`

[4]`https://www.paddlepaddle.org.cn/en/`

Koh & Liang (2017) investigated the effect of error-maximizing noise on models. Huang et al. (2021) proposed the error-minimizing noise based on a assumption that reduction of training errors (close to zero) can trick the model into believing there is nothing to learn from unlearnable examples. Inspired by the idea that adversarial noise is a type of non-robust features (Ilyas et al., 2019), Fowl et al. (2021) proposed using targeted adversarial examples to implement poisoning attacks called adversarial poisoning. Yu et al. (2022) proposed synthetic perturbations by enabling the model to rely on shortcuts (Geirhos et al., 2020) during the training phase.

Adversarial Attacks. Adversarial attacks explore the vulnerability of DNNs, where they catastrophically misclassify inputs by small perturbations (Szegedy et al., 2013). However, regarding it as a remarkable tool to interfere algorithms, many efforts have studied privacy preserving in testing phase (Zhang et al., 2020; Zhong & Deng, 2022). Recently, many works have also noticed this property of adversarial attacks when protecting privacy in the training phase and have used its gradient-based generation paradigm to construct powerful poisoning attacks (unlearnable examples) (Huang et al., 2021; Fowl et al., 2021). However, regardless of whether the image noises are the sample-wise or the class-wise (Moosavi-Dezfooli et al., 2017), they rely on labelled information and miss the label-inconsistency issue.

3 Problem Statement

3.1 Preliminaries

Threat Models. We introduce two parties: the *protector*, and the *hacker*. The protector possesses the *surrogate model* and uses it to implement a poisoning attack to release the generated unlearnable examples onto the web. In the real world, it is close to social media companies, who have the drive to take steps to protect users' privacy. Then, without permission, the hacker crawls the training images they need from the web, and in some way label them as supervision to train the *victim model* for private purposes. Labelling images may be done through their own manuals or by using crowd-sourcing platforms, such as Amazon Mechanical Turk. In the real world, a hacker may be a researcher for academic purposes or a practitioner in industry.

Assumptions and Objectives. We consider image classification task in this paper. We first briefly review the pipeline of previous studies on this task. Given a clean training dataset $\mathcal{D}_c^m = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=j}^k$ consisting of k clean examples with images $\boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^d$ and labels $\boldsymbol{y} \in \mathcal{Y}$ and the total number of classes m, the protector trained the m-class surrogate model f_s^m on it. Next, based on the clean training dataset $\mathcal{D}_c^m$ and the corresponding surrogate model f_s^m, the protector generates the unlearnable dataset $\mathcal{D}_u^n = \{(\boldsymbol{x}_i', \boldsymbol{y}_i')\}_{i=1}^k$ with unlearnable images $\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{\delta}$ and $\boldsymbol{x} \in \mathcal{D}_c^m$ and imperceptible noise $\boldsymbol{\delta} \in \Delta \subset \mathbb{R}^d$ and labels $\boldsymbol{y}' \in \Omega$ and the total number of classes n. Finally, the protectors expect that the hackers will use this unlearnable dataset $\mathcal{D}_u^n$ with the same labels to train their own n-class victim model f_v^n with the same classes as the surrogate model f_s^m, i.e., their assumption is that $\boldsymbol{y} = \boldsymbol{y}'$, $m = n$, and $\mathcal{Y} = \Omega$. They ignore the fact that the "protector$\rightarrow$ web$\rightarrow$ hacker" transfer chain of unlearnable dataset involves only images but not labels. So this assumption is nearly impossible in reality, since it is hard for protectors and hackers to label all images identically. We denote the above assumption as label-consistency.

In this paragraph, we will clarify the more practical assumption called label-inconsistency. Still considering the clean training dataset $\mathcal{D}_c^m$ on the classification task, a more reasonable surrogate model is supposed to be the classified model $f_s^{\overline{m}}$ that has seen as many classes of images as possible, where $\overline{m}$ is not equal to m, and usually $\gg m$. In this paper, we employ ResNet-50 (He et al., 2016) trained on ImageNet-1k as the default surrogate model, i.e., $\overline{m} = 1000$. We also discuss other kinds of surrogate models later. Next, the unlearnable examples $\boldsymbol{x}'$ are generated from the data pairs $(\boldsymbol{x}, \overline{\boldsymbol{y}})$ fed into the surrogate model $f_s^{\overline{m}}$, where $\overline{\boldsymbol{y}}$ is the pseudo label predicted by the surrogate model $f_s^{\overline{m}}$, i.e., $\overline{\boldsymbol{y}} = f_s^{\overline{m}}(\boldsymbol{x})$. Finally, hackers use these unlearnable sample pairs $(\boldsymbol{x}', \boldsymbol{y}')$ to train their own n-class victim model f_v^n. Note that although $\boldsymbol{y}'$ and $\overline{\boldsymbol{y}}$ are most likely not equal, they are similar in semantics.

Our aim is to shed the limitation on the number of categories regardless of m or n, to generate the perturbation $\boldsymbol{\delta}$ and the corresponding unlearnable examples $\boldsymbol{x}'$ that are also applicable in the label-inconsistency scenario. Regarding the non-classification surrogate model f_s (or the classification

surrogate model but removed the top fully-connected layers), we pursue to perturb the embedding features $\boldsymbol{e} = f_s(\boldsymbol{x})$ to $\boldsymbol{e}' = f_s(\boldsymbol{x}')$ (instead of focusing the change of prediction) by adding perturbation $\boldsymbol{\delta}$.

3.2 Baseline Results in Label-inconsistency

In this subsection, we probe the performance of the current baseline methods in the label-inconsistency scenario. Taking the CIFAR-10 dataset (Krizhevsky et al., 2009) as an example, we first evaluate the performance of the baseline methods under label-consistency. Under this ideal assumption, the labels for generating unlearnable examples are consistent with the labels for training the victim model. Next, following the label-inconsistency scenario presented in the previous subsection, we employ ResNet-50 (trained on ImageNet-1k) as the surrogate model to generate the unlearnable examples. For the same image, the label at the generation of unlearnable examples stage and at the training of victim model stage may be inconsistent, e.g., "partridge" at the first stage and "bird" at the second stage. Even the 5000 "bird" images in the second stage belong to dozens of different categories in the first stage.

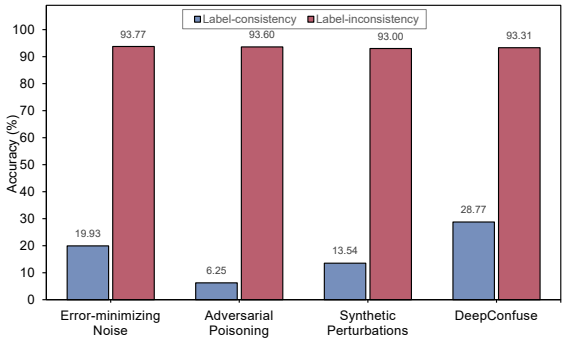
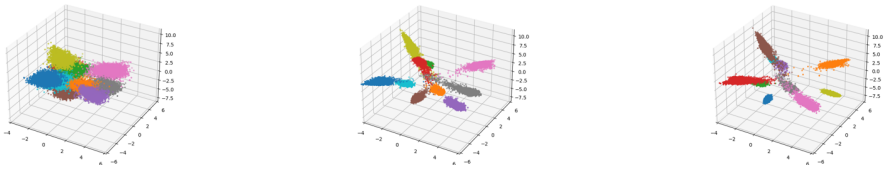


Figure 1: CIFAR-10 accuracy of the victim models derived by the current SOTA attacks in different scenarios.

In this setting, we compare the performance of baseline methods including Error-minimizing Noise (Huang et al., 2021), Error-maximizing Noise (Koh & Liang, 2017), Adversarial Poisoning (Fowl et al., 2021), Synthetic Perturbations (Yu et al., 2022) and DeepConfuse (Feng et al., 2019) in these two scenarios. We showed in Figure 1 that these SOTA attacks are rendered inefficient in the label-inconsistency scenario even though they yield considerable attack performance in their raw assumptions (label-consistency).

4 Universal Unlearnable Examples: Cluster-wise Perturbations

4.1 Uniformity and Discrepancy



(a) clean (b) Error-minimizing Noise (c) Adversarial Poisoning

Figure 2: The three-dimensional feature visualization of (a) clean CIFAR-10 examples and (b) (c) the unlearnable examples derived by poisoning attacks. Data with the same color indicate that they belong to the same category.

In the previous section, we showed that the gradient-based methods (Error-minimizing Noise and Adversarial Poisoning) can achieve comparable results in the label-consistency scenario actually. This demonstrates that the drop in effectiveness is due to a lack of adaptation to the new label-inconsistency scenario, rather than the erroneous nature in the methodology. Therefore, a more intuitive solution is to find ways to decouple these attacks from the label dependencies to fit the label-inconsistency scenario. To this end, we analyzed the changes caused by the perturbations on embedding representations, but with less concern for the final output of the model. By this, we sought to find the common core mechanisms behind them, even without the labelled information.

Specifically, using a ResNet-18 trained on the CIFAR-10 dataset as a mapping function, a 512-dimensional representation is derived for each image in CIFAR-10 dataset with an input resolution of $32 \times 32 \times 3$. To be more intuitive, we used the representation matrix of the original CIFAR-10 to construct a three-dimensional space by PCA decomposition (Wold et al., 1987) (512-dimension to 3-dimension). The visualizations of the clean examples, the unlearnable examples generated by Error-minimizing Noise and Adversarial Poisoning in this three-dimensional space are plotted in the Figure 2, respectively. (1) Error-minimizing Noise causes samples belonging to the same class to be more compact in feature space, but does not shift the data manifold. This actually disrupts the discrepancy of the data distribution. In other words, e.g., for the category "birds", even if there are 5000 different "birds" in the training set, they all look like the same "bird" to the model. (2) On top of disrupting the discrepancy, Adversarial Poisoning also shifts the data manifold away from the original region to disrupt the uniformity simultaneously. Also using the category of "birds" as an example, these 5000 images not only all look the same, but also no longer looked like "birds" to the model. So, once we can achieve disrupting the uniformity and discrepancy without relying on labelled information, then theoretically, we can also enable effective unlearnable examples in the label-inconsistency scenario.

4.2 Class-wise Perturbations and Cluster-wise Perturbations

Class-wise Perturbations. The previous analytical results have demonstrated that adversarial perturbation is a remarkable solution for disrupting the uniformity and discrepancy. The adversarial perturbation can be further divided into two categories: class-wise perturbation and sample-wise perturbation. Different from general sample-wise perturbation, class-wise perturbation is also known as universal adversarial perturbation (Moosavi-Dezfooli et al., 2017; Poursaeed et al., 2018), which is fixed image noise that can be added into arbitrary images to fool the model. Huang et al. (2021) have shown that the class-wise error-minimizing noise is superior in terms of experimental results. We provide some insights into the class-wise perturbation here. First, Zhang et al. (2021) discovered that, when investigating the feature preferences of DNNs, the class-wise perturbation can be regarded as the "strong features" that effectively cover native semantic features in images. Such strong features cause images to be more clustered on the feature space, which actually enhances the disrupting discrepancy. Second, Yu et al. (2022) found that, when conducting backdoor attacks, the fixed pattern (shortcuts) is an effective and model-unrequired unlearnable noise. Therefore, the complementary properties give the class-wise perturbation a boost in performance.

However, the limitation of the class-wise perturbation is also obvious as it requires labelled information, so it is not appropriate for the label-inconsistency scenario. Besides, generating a single class-wise perturbation is quite time consuming (requiring several epochs on the entire training set). If considering the surrogate model possesses a huge number of classes (e.g., 1000 categories for ImageNet-1k models), the huge computational cost is unimaginable.

Cluster-wise Perturbations. To solve this limitation, we propose a universal cluster-wise perturbations called *UniversalCP* that does not require labelled information yet achieves disrupting the uniformity and discrepancy. We use the generators (encoder-decoder networks (Poursaeed et al., 2018)) to yield cluster-wise perturbation. The whole pipeline can be implemented in two parts. First, the clean dataset $\mathcal{D}_c$ is fed into the surrogate model without classified layers f_s to derive the feature matrix $\mathbb{E} = [\boldsymbol{e}_1, \cdots, \boldsymbol{e}_k]$. Through feeding the it into *K*-means algorithm (Selim & Ismail, 1984), given the number of clusters p, the set of clusters $\mathcal{C} = \{\mathcal{C}_1, \cdots, \mathcal{C}_p\}$ where $\mathcal{C}_i = \{\boldsymbol{x}_{ij}\}_{j=1}^{\tau(i)} = \{\boldsymbol{x}_{i1}, \cdots, \boldsymbol{x}_{i\tau(i)}\}$ and $\sum_{i=1}^{p} \tau(i) = k$, and the corresponding cluster centers $\mu_{\mathcal{C}} = \{\mu_{\mathcal{C}_1}, \cdots, \mu_{\mathcal{C}_p}\}$ are derived. Second, we generate the corresponding cluster-wise perturbation $\boldsymbol{\delta}_i$ for each cluster $\mathcal{C}_i$. Specially, for i-th cluster $\mathcal{C}_i$, we hope that $\boldsymbol{\delta}_i$ will bring the cluster of unlearnable examples $\mathcal{C}_i' = \{\boldsymbol{x}_{i1}', \cdots, \boldsymbol{x}_{i\tau(i)}'\}$ close to one of the other clustering center $g(\mu_{\mathcal{C}_i})$ in the feature space of the surrogate model f_s:

$$\begin{aligned} \theta_i &= \arg\min_{\theta_i} \mathcal{L}_{\mathrm{DDU}}(\mathcal{C}_i, g(\mu_{\mathcal{C}_i}), \theta_i) \\ &= \arg\min_{\theta_i} \sum_{\boldsymbol{x}_i j \in \mathcal{C}_i} d(f_s(\boldsymbol{x}_i j + \mathcal{G}(\sigma; \theta_i)), g(\mu_{\mathcal{C}_i})), \end{aligned} \tag{1}$$

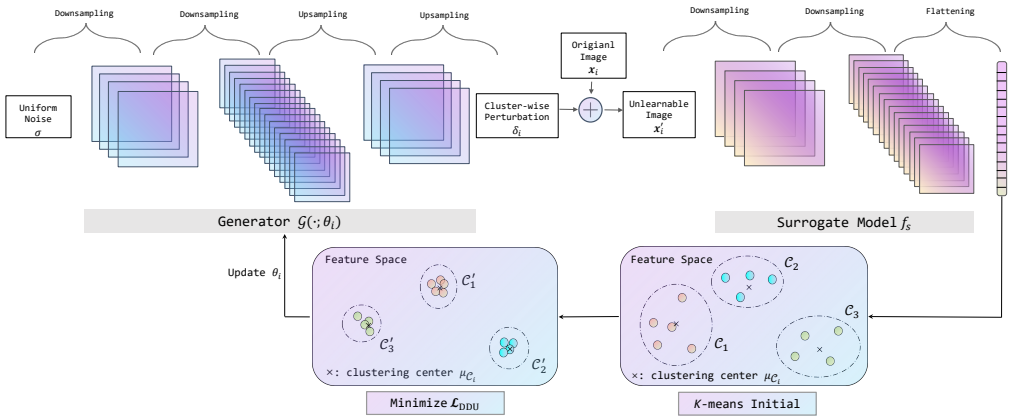


Figure 3: The UniversalCP pipeline. The entire dataset is divided into p clusters, each corresponding to a certain generator parameters θ_i and a cluster-wise perturbation $\boldsymbol{\delta}_i$.

where $\mathcal{L}_{\text{DDU}}$ denotes the loss of disrupting discrepancy and uniformity. $\mathcal{G}$ is the generator with parameters θ_i, σ is the uniform noise sampled from a uniform distribution $[0, 1]$. $d(\cdot, \cdot)$ is a distance metric, and Kullback-Leibler divergence loss is employed in this work. g is a permutation on $\mu_{\mathcal{C}}$ to ensure the disrupting uniformity, i.e., $g(\mu_{\mathcal{C}_i}) \neq \mu_{\mathcal{C}_i}$. When the generator $\mathcal{G}$ with parameters θ_i is optimized by Equation 1, then the cluster-wise perturbation $\boldsymbol{\delta}_i$ can be obtained by the following equation:

$$\boldsymbol{\delta}_i = \mathcal{G}(\sigma; \theta_i). \tag{2}$$

Equation 1 and Equation 2 need to be repeated p times to obtain cluster-wise perturbations $\boldsymbol{\delta} = \{\boldsymbol{\delta}_1, \cdots, \boldsymbol{\delta}_p\}$. Figure 3 illustrates the generation of the cluster-wise perturbations. The detailed pipeline is described in Algorithm 1.

4.3 Surrogate Model: CLIP

How to choose a surrogate model remains a challenge to be discovered. The knowledgeable surrogate models can generate better unlearnable examples on different datasets and victim models, which is analogous to "transferability" in the adversarial examples community. Since the baseline methods use a classification model as a surrogate model, an intuitive choice would be a model trained on a large-scale classification dataset such as ImageNet. Benefit from the removal of the limitation on the number of categories, we can extend the choice of surrogate models to numerous non-classified deep learning models. Among them, the size of the training set used by vision-and-language pre-training models far exceeds the existing classification dataset, e.g., CLIP uses 400 million images far beyond ImageNet. Beyond that, CLIP uses text information as supervision rather than one-hot labels. This more semantical and informative supervision allows the model to learn more general knowledge (better transferability), as evidenced by the SOTA transfer results on more than 30 downstream computer vision datasets (Radford et al., 2021). These two items motivate us to use CLIP as a surrogate model to generate more transferable perturbations.

5 Experiments

5.1 Experimental Settings

In this paper, we carry out our study on 6 high-resolution and industrial-scale vision datasets to simulate real-world performance including Pets (Parkhi et al., 2012), Cars (Krause et al., 2013), Flowers (Nilsback & Zisserman, 2008), Food (Bossard et al., 2014), SUN397 (Xiao et al., 2010) and ImageNet (Russakovsky et al., 2015). For ImageNet, we use the first 100 classes subset denoted as ImageNet*. For surrogate models, we consider ResNet-50 (RN50) trained on ImageNet-1k as the default, unless otherwise explicitly stated. In Section 5.3, we also discuss the performance of ViT (Dosovitskiy et al., 2020). For victim models, we employ randomly initialized ResNet-18 (RN18) (He et al., 2016), EfficientNet-B1 (EN1) (Tan & Le, 2019) and RegNetX-1.6GF (RNX1.6) (Radosavovic

et al., 2020). We consider L_∞-norm restriction in this work, within $\|\boldsymbol{\delta}\|_\infty < \epsilon = 16/255$. The number of clusters p is a hyperparameter defaulted to 10 in this work, and we discuss its effect on the experimental results in Section 5.5.

We compare our methods with baseline methods including DeepConfuse (Feng et al., 2019), Synthetic Perturbations (Yu et al., 2022), Error-minimizing Noise (Huang et al., 2021), Error-maximizing Noise (Koh & Liang, 2017), and Adversarial Poisoning (Fowl et al., 2021). We also extend the Error-minimizing Noise and Adversarial Poisoning to transform the targets from labels to cluster centers, denoted as Error-minimizing Noise-C and Adversarial Poisoning-C, respectively. More setups are detailed in Appendix B.

5.2 Experimental Results on Black-box Victim Models

Table 1: The test accuracy (%) of the victim models RN18 trained on different datasets and attack methods, all of which use RN50 as the backbone of the surrogate model.

Methods	Pets	Cars	Flowers	Food	SUN397	ImageNet*
Clean	62.31	67.18	67.18	78.97	43.08	77.76
DeepConfuse	53.72	51.11	50.94	73.13	34.41	55.12
Synthetic Perturbation	52.60	53.50	52.74	74.80	38.26	74.69
Error-maximizing Noise	54.70	52.95	51.70	73.77	37.57	73.82
Error-minimizing Noise	52.96	54.43	50.58	75.47	38.48	74.20
Error-minimizing Noise-C	54.54	50.93	51.49	74.91	38.20	73.56
Adversarial Poisoning	50.86	51.91	50.64	75.07	38.51	73.76
Adversarial Poisoning-C	53.12	53.20	52.40	74.97	37.70	73.60
UniversalCP (Ours)	**7.41**	**35.84**	**41.86**	**55.29**	**20.38**	**54.80**
UniversalCP-CLIP (Ours)	**4.69**	**16.70**	**7.97**	**19.07**	**3.89**	**39.78**

In this subsection, we compare our methods with baseline methods on several black-box victim models. We conduct the experiment on ResNet-18, EfficientNet-B1 and RegNetX-1.6GF, and the results are shown in Table 1, Table 4 and Table 5, respectively. The unlearnable images input to each victim model are subjected to the following data augmentations: resizing, random crop, random horizontal flip and normalization. Since we do not consider the data augmentations, the common setup of training DNNs in real-world, when deploying poisoning attacks, they actually weaken the effect of unlearnable examples. From Table 1, we have the following main findings: (1) Our methods, UniversalCP and UniversalCP-CLIP, widely outperform these existing methods. This demonstrates the superiority of our method in label-inconsistency scenario compared to the near invalidation of the baseline methods. (2) UniversalCP-CLIP achieves a better performance than UniversalCP, which proves that using CLIP as the surrogate model is a reasonable and compatible way. (3) For a specific case, ImageNet*, the victim model training set is a subset of the surrogate model training set, where our methods still achieve the best performance. This further demonstrates the effectiveness of using CLIP, as the similarity of ImageNet* to ImageNet is much higher than the CLIP's training set.

5.3 Experimental Results on Black-box Commercial Platforms

We have demonstrated the effectiveness of our methods on black-box victim models. To further verify our methods in real world, we deploy two commercial machine learning platforms: Microsoft `Azure` and Baidu `PaddlePaddle`. In this setting, all the training details are owned to the platforms and are unknown to us, including the model, learning rate, batch size, epoch, data augmentation, splitting of the validation set, etc. Hence, the evaluation results on these two APIs are even more valuable than Table 1. Considering that ViT may be used on commercial platforms due to its recent popularity, we extended UniversalCP-CLIP-ViT from the setup in Section 5.2, i.e., replaced the surrogate model ResNet-50 in UniversalCP-CLIP with ViT-B-32. The results in Table 2 are consistent with Table 1, i.e., our methods achieve the best performance and can be productively complemented with CLIP. Moreover, UniversalCP-CLIP-ViT performs better than UniversalCP-CLIP-RN50 on both platforms, suggesting that ViT may be a better choice for surrogate model if the victim model is deployed by commercial platforms. We visualize some examples in Figure 6.

Table 2: The test accuracy (%) of Cars dataset on commercial platforms. For both platforms, we used the fastest training configuration.

METHODS	Azure	PaddlePaddle
CLEAN	48.45	83.74
DEEPCONFUSE	39.47	41.88
SYNTHETIC PERTURBATION	42.38	47.59
ERROR-MAXIMIZING NOISE	42.83	42.99
ERROR-MINIMIZING NOISE	44.06	44.40
ERROR-MINIMIZING NOISE-C	45.84	44.53
ADVERSARIAL POISONING	43.97	43.38
ADVERSARIAL POISONING-C	46.03	45.10
UNIVERSALCP-RN50 (OURS)	**36.4**	**30.96**
UNIVERSALCP-CLIP-RN50 (OURS)	**26.97**	**25.79**
UNIVERSALCP-CLIP-VITB32 (OURS)	**22.47**	**11.49**

5.4 EXPERIMENTAL RESULTS ON DEFENSES

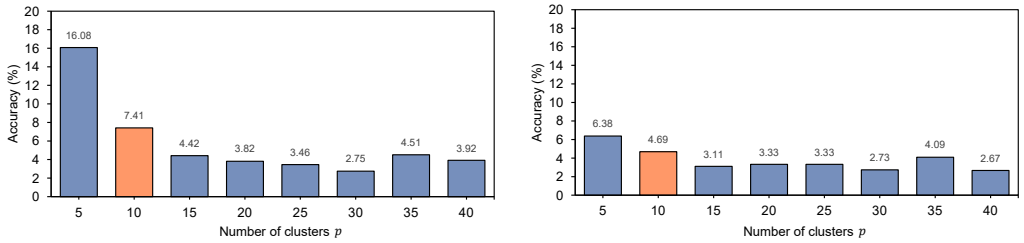
Table 3: Effects of defenses against our attacks on Pets dataset. Each defense is incorporated with data augmentation.

METHODS	NO DEFENSE	MIXUP	GAUSSIAN	CUTMIX	CUTOUT
UNIVERSALCP (OURS)	7.41	14.34	24.26	14.50	12.35
UNIVERSALCP-CLIP (OURS)	4.69	11.96	18.59	6.21	12.29

Since the poisoning attack derives the invisible perturbations to degrade the performance of the victim models, it is worth exploring the impact on the image pre-processing. This image pre-processing aimed at weakening poisoning attacks can also be referred to as "defense". Previous studies have reported that adversarial training is one of the strongest of these defenses (Madry et al., 2017; Fu et al., 2022). However, for the research area of using invisible perturbation to protect user privacy on the web, it is paranoid to consider adversarial training as a potential defense. The reason is as follows: images on the social web are vast in number, common in kind, and widely available, so hackers use this cheap way to obtain the required training data. But once adversarial training is deployed, then it becomes very expensive considering that the cost of adversarial training is one or two order of magnitude higher than normal training, regardless of the time cost or the computational cost on the GPUs. In other words, unless it is a rare type of image like medical images, instead of using adversarial training to deal with unlearnable examples, hackers could just purchase certified data. Therefore, a more practical defense should be fast, such as some image denoising techniques. The regular data augmentation we used in Section 5.2 can also be considered as a simple defense. In this subsection, we also include others techniques such as Mixup (Zhang et al., 2017), Gaussian smoothing, Cutmix (Yun et al., 2019) and Cutout (Cubuk et al., 2018) as defenses, and the defense results on RN18 are shown in Table 3. The experimental results show that these defenses even combined with regular data augmentation are ineffective against our methods, but can hurt the performance of victim model without defenses (e.g., Gaussian smoothing).

5.5 ABLATION STUDY

We have shown that the SOTA results are achieved by our methods at $p = 10$, but the influence of hyperparameter p has not yet been explored. Taking the Pets dataset as an example, we evaluated UniversalCP and UniversalCP-CLIP under different values of p as shown in Figure 4. First, we can find that our methods achieve reasonable results for all values of p, and even the worst value ($p = 5$) performs better than baseline methods. This further illustrates the independent of our methods to delicate hyperparameters. Moreover, for this dataset with 37 classes, $p = 10$ is chosen casually rather than carefully, suggesting that the results in Table 1 have the potential to achieve better results if the hyperparameter p is carefully tuned.



(a) UniversalCP (b) UniversalCP-CLIP

Figure 4: Effects of the number of cluster p on Pets dataset. $p = 10$ is used as default.

5.6 Mixture of Clean Data and Unlearnable Data

So far, the experiments conducted have been in the setting of entire unlearnable training dataset, consistent with Huang et al. (2021); Fowl et al. (2021); Yu et al. (2022). This setting is reasonable when the protectors have access to modify all user data to prevent hackers from malicious collections, such as social media platforms. However, a further case is that the hacker's property possesses some unperturbed images, which creates a mixture of clean data and unlearnable data. This case was shown by previous studies (Huang et al., 2021; Fowl et al., 2021; Yu et al., 2022) that the effectiveness of unlearnable samples is severely weakened, where a mixture of unlearnable data and clean data yielded even higher accuracy than only clean data, i.e., the unlearnable data played a negative role.

Figure 5: Effects of the mixture of unlearnable data and clean data. The analysis was carried out on the Pets dataset.

In this work, we compared using a portion of clean data as the training set and using the same clean data supplemented with unlearnable data as the training set. E.g., for Pets dataset containing 3680 training images of 37 classes, when we use only 2 classes of clean data to build the "clean" training set, we also use these 2 classes of clean data and the remaining 35 classes of unlearnable data to build the "mixture" training set. As shown in Figure 5, the purple line is near the green line, indicating that the additional unlearnable data is almost useless for the model's test performance, where each green dot indicates that the all 3680 training images were used. We can also find that when the number of classes used is small, the accuracy derived from the mixture dataset is slightly higher than the clean dataset, which is consistent with the previous finding (Huang et al., 2021; Fowl et al., 2021; Yu et al., 2022). However, when the number of classes is relatively high (roughly ≥ 19), the mixture dataset is almost equal to or sometimes slightly lower than the clean dataset, which is an undiscovered result in previous line of work. Still, the mixture case is expected to perform as well as the entire training dataset in the future.

6 Conclusion

Unlearnable examples show acceptable potential in preventing hackers from collecting users' private information on the Internet. Several works have noticed such paths and proposed effective poisoning attacks under the ideal assumption. However, in the present work, we clarify a more realistic scenario called label-inconsistency which weakens the effectiveness of existing methods. By analyzing the common mechanisms of existing attacks, we propose the cluster-wise perturbations for the more challenging scenario. We also investigate the criteria for selecting surrogate models. The experimental

results on a variety of settings demonstrate the effectiveness of our cluster-wise perturbations and the strategy for selecting surrogate models.

REFERENCES

Battista Biggio and Fabio Roli. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 84:317–331, 2018.

Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. *arXiv preprint arXiv:1206.6389*, 2012.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European conference on computer vision*, pp. 446–461. Springer, 2014.

Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Ji Feng, Qi-Zhi Cai, and Zhi-Hua Zhou. Learning to confuse: generating training time adversarial data with auto-encoder. *Advances in Neural Information Processing Systems*, 32, 2019.

Liam Fowl, Micah Goldblum, Ping-yeh Chiang, Jonas Geiping, Wojciech Czaja, and Tom Goldstein. Adversarial examples make strong poisons. *Advances in Neural Information Processing Systems*, 34:30339–30351, 2021.

Shaopeng Fu, Fengxiang He, Yang Liu, Li Shen, and Dacheng Tao. Robust unlearnable examples: Protecting data against adversarial learning. *arXiv preprint arXiv:2203.14533*, 2022.

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.

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

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Kashmir Hill. The secretive company that might end privacy as we know it. In *Ethics of Data and Analytics*, pp. 170–177. Auerbach Publications, 2020.

Hanxun Huang, Xingjun Ma, Sarah Monazam Erfani, James Bailey, and Yisen Wang. Unlearnable examples: Making personal data unexploitable. In *International Conference on Learning Representations*, 2021.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *Advances in neural information processing systems*, 32, 2019.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pp. 1885–1894. PMLR, 2017.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pp. 554–561, 2013.

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

Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34:9694–9705, 2021.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1765–1773, 2017.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pp. 722–729. IEEE, 2008.

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pp. 3498–3505. IEEE, 2012.

Omid Poursaeed, Isay Katsman, Bicheng Gao, and Serge Belongie. Generative adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4422–4431, 2018.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.

Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10428–10436, 2020.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

Shokri Z Selim and Mohamed A Ismail. K-means-type algorithms: A generalized convergence theorem and characterization of local optimality. *IEEE Transactions on pattern analysis and machine intelligence*, (1):81–87, 1984.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pp. 6105–6114. PMLR, 2019.

Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pp. 3485–3492. IEEE, 2010.

Da Yu, Huishuai Zhang, Wei Chen, Jian Yin, and Tie-Yan Liu. Availability attacks create shortcuts. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2367–2376, 2022.

Chia-Hung Yuan and Shan-Hung Wu. Neural tangent generalization attacks. In *International Conference on Machine Learning*, pp. 12230–12240. PMLR, 2021.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6023–6032, 2019.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Jiaming Zhang, Jitao Sang, Xian Zhao, Xiaowen Huang, Yanfeng Sun, and Yongli Hu. Adversarial privacy-preserving filter. In *Proceedings of the 28th ACM International Conference on Multimedia*, pp. 1423–1431, 2020.

Jiaming Zhang, Jitao Sang, Qi Yi, Yunfan Yang, Huiwen Dong, and Jian Yu. Pre-training also transfers non-robustness. *arXiv preprint arXiv:2106.10989*, 2021.

Yaoyao Zhong and Weihong Deng. Opom: Customized invisible cloak towards face privacy protection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

A ALGORITHM OF CLUSTER-WISE PERTURBATIONS

Algorithm 1 Generating Cluster-wise Perturbations

Input: surrogate model f_s, distance metric d, uniform noise σ, number of clusters p, random permutation g, L_∞-norm restriction ϵ, clean images $\boldsymbol{x} \in D_c$, initialized generator $\mathcal{G}$ with parameters θ
Output: cluster-wise perturbations $\boldsymbol{\delta} = \{\boldsymbol{\delta}_1, \cdots, \boldsymbol{\delta}_p\}$
feature matrix $\mathbb{E} = f_s(\boldsymbol{x})$
set of clusters with the cluster centers $\{\mathcal{C}, \mu_{\mathcal{C}}\} = K\text{-means}(\mathbb{E}, p)$
for i **in** $1 \cdots p$ **do**
    Initialize θ_i
    $\boldsymbol{\delta}_i = \mathcal{G}(\sigma; \theta_i)$
    $\boldsymbol{\delta}_i = \text{Clamp}(\boldsymbol{\delta}_i, -\epsilon, \epsilon)$
    for $\boldsymbol{x}_{ij}$ **in** $\mathcal{C}_i$ **do**
        $\boldsymbol{x}'_{ij} = \text{Clamp}(\boldsymbol{x}_{ij} + \boldsymbol{\delta}_i, 0, 1)$
        $\theta_i \leftarrow \text{Optimize}(\boldsymbol{x}'_{ij}, f_s, g(\mu_{\mathcal{C}_i}), d)$
    end for
    $\boldsymbol{\delta}_i = \mathcal{G}(\sigma; \theta_i)$
    $\boldsymbol{\delta}_i = \text{Clamp}(\boldsymbol{\delta}_i, -\epsilon, \epsilon)$
end for

B MORE EXPERIMENTAL DETAILS

All analyses and experiments are conducted on NVIDIA Tesla V100 32GB GPUs. The code for the analyses and experiments uses PyTorch framework.

For ImageNet*, we repeated p times to train the generator $\mathcal{G}$ for 10 epochs with SGD using the initial learning rate 0.1 and Cosine annealing, 256 batch size, and momentum 0.9. For others datasets, we train the generator $\mathcal{G}$ for 50 epochs. For random permutation g, we simply chose $i \rightarrow i + 1$ to build a closed loop. For the surrogate models, we load the weights of ResNet-50 from the torchvision package, as well as the weights of CLIP released by the authors. According to Radford et al. (2021), there are 3 minor changes for ResNet-50 of CLIP: (1) There are now 3 "stem" convolutions as opposed to 1, with an average pool instead of a max pool. (2) Performs anti-aliasing strided convolutions, where an avgpool is prepended to convolutions with stride > 1. (3) The final pooling layer is a QKV attention instead of an average pool. We train all randomly initialized victim models including ResNet-18, EfficientNet-B1 and RegNetX-1.6GF for 90 epochs with SGD using the initial learning rate 0.1 and Cosine annealing, 256 batch size, and momentum 0.9. For adversarial attacks in Error-minimizing Noise and Adversarial Poisoning, we use PGD-40, and the step size is set to 1.25.

C MORE EXPERIMENTAL RESULTS ON BLACK-BOX VICTIM MODELS

Table 4: The test accuracy (%) of the victim models EfficientNet-B1 trained on different datasets and attack methods, all of which use RN50 as the backbone of the surrogate model.

METHODS	PETS	CARS	FLOWERS	FOOD	SUN397	IMAGENET*
CLEAN	48.68	72.33	52.46	80.29	42.84	78.04
DEEPCONFUSE	35.54	47.15	43.28	72.91	35.22	45.74
SYNTHETIC PERTURBATION	28.02	58.34	42.93	74.99	35.92	72.94
ERROR-MAXIMIZING NOISE	33.71	55.64	42.66	74.40	37.30	73.72
ERROR-MINIMIZING NOISE	36.88	54.23	44.06	75.54	37.20	72.20
ERROR-MINIMIZING NOISE-C	33.85	50.21	45.86	75.07	37.80	72.06
ADVERSARIAL POISONING	37.99	50.08	**41.65**	74.88	36.44	72.54
ADVERSARIAL POISONING-C	25.32	48.59	42.84	73.92	36.89	71.82
UNIVERSALCP (OURS)	**4.85**	**46.54**	**41.78**	**53.45**	**22.97**	**32.30**
UNIVERSALCP-CLIP (OURS)	**6.19**	**1.43**	**17.61**	**17.44**	**12.95**	**31.82**

Table 5: The test accuracy (%) of the victim models RegNetX-1.6GF trained on different datasets and attack methods, all of which use RN50 as the backbone of the surrogate model.

METHODS	PETS	CARS	FLOWERS	FOOD	SUN397	IMAGENET*
CLEAN	44.86	63.84	52.69	84.02	43.27	80.78
DEEPCONFUSE	33.71	41.15	46.01	77.26	33.52	49.88
SYNTHETIC PERTURBATION	34.51	45.54	47.16	77.65	37.78	60.38
ERROR-MAXIMIZING NOISE	34.26	43.40	46.25	–	37.82	76.72
ERROR-MINIMIZING NOISE	37.04	39.67	47.34	79.43	36.82	74.86
ERROR-MINIMIZING NOISE-C	34.15	44.19	47.81	78.73	37.86	75.86
ADVERSARIAL POISONING	34.29	46.06	47.41	78.64	36.42	76.32
ADVERSARIAL POISONING-C	33.50	45.14	46.20	79.04	36.62	76.60
UNIVERSALCP (OURS)	**4.77**	**29.46**	**37.97**	–	**22.28**	**56.10**
UNIVERSALCP-CLIP (OURS)	**3.87**	**1.46**	**7.51**	–	**6.04**	**41.66**

We compare our methods with baseline methods on EfficientNet-B1 and RegNetX-1.6GF shown in Table 4 and Table 5, respectively.

D EXAMPLE GALLERY

(a) UNIVERSALCP-RN50

(b) UNIVERSALCP-CLIP-RN50

(a) UNIVERSALCP-CLIP-VITB32

Figure 6: Samples from perturbed images uploaded to commercial platforms under $\epsilon = 16/255$.

In Figure 6, we show some examples of perturbed images uploaded to commercial platforms Microsoft `Azure` and Baidu `PaddlePaddle`.