ADVERSARIAL ATTACK ACROSS DATASETS

Anonymous authors
Paper under double-blind review

ABSTRACT

It has been observed that Deep Neural Networks (DNNs) are vulnerable to transfer attacks in the query-free black-box setting. However, the previous studies on transfer attack commonly assume that the white-box surrogate models possessed by the attacker and the black-box victim models are trained on the same dataset, which means the attacker implicitly knows the label set and the input size of the victim model. However, this assumption is usually unrealistic as the attacker may not know the dataset used by the victim model, and further, the attacker needs to attack any randomly encountered images that may not come from the same dataset. Therefore, in this paper we define a new Generalized Transferable Attack (GTA) problem where we assume the attacker has a set of surrogate models trained on different datasets (with different label sets and image sizes), and none of them is equal to the dataset used by the victim model. We then propose a novel method called Image Classification Eraser (ICE) to erase classification information for any encountered images from arbitrary dataset. Extensive experiments on Cifar-10, Cifar-100, and TieredImageNet demonstrate the effectiveness of the proposed ICE on the GTA problem. Furthermore, we show that existing transfer attack methods can be modified to tackle the GTA problem, but with significantly worse performance compared with ICE.

1 INTRODUCTION

It has been observed that by adding human imperceptible adversarial perturbations to clean input data, even well-trained deep neural networks (DNNs) can be fooled with a high probability (Szegedy et al., 2014; Goodfellow et al., 2014; Carlini & Wagner, 2017; Lin et al., 2020). As a result, the security and robustness of DNNs have attracted growing attention from both academia and industry (Uesato et al., 2018; Croce & Hein, 2020b; Sriramanan et al., 2020; Haizhong et al., 2020). Existing methods for generating adversarial examples, also known as "attacks", can be categorized by the following different threat models: white-box (Goodfellow et al., 2014; Moosavi-Dezfooli et al., 2016), query-based black-box (Brendel et al., 2018; Ilyas et al., 2018; Cheng et al., 2018), and query-free black-box attack (Liu et al., 2017; Papernot et al., 2017). As Table 1 shows, in the white-box attack setting, the attacker can access all information of the victim model while in query-based or query-free black-box setting, the victim model is hidden from the attacker.

In this paper, we consider the query-free black-box setting, where the victim model is black-box and no queries can be made (Wu et al., 2020c; Wang & He, 2021). Even in this restricted setting, it has been shown that DNN models are still vulnerable due to the existence of transfer attacks (Wu et al., 2018; 2020c; Naseer et al., 2019; Demontis et al., 2019), which leverage one or a few surrogate white-box models to construct adversarial examples and transfer them to the victim model. Despite of the fact that previous works have demonstrated the effectiveness of transfer attacks (Papernot et al., 2017; Guo et al., 2020) and much efforts have been made recently to improve transfer attacks (Li et al., 2020c; Wang & He, 2021), they commonly made an implicit assumption that **surrogate models and the victim model are trained on the same dataset**, and the successful results commonly rely on this assumption. This means the attacker knows the **input resolution** and **label set** of the victim model. For example, when the victim model and images are from Cifar-10 (Krizhevsky et al., 2009), they assume the surrogate model is also trained on Cifar-10 instead of ImageNet (Deng et al., 2009).

However, in practical situations, the test image (and victim model) can come from any dataset; the attackers won't know which dataset is being used and it is unlikely to retrain a new surrogate model for each new dataset. To tackle this more practical setting, we need to assume the surrogate models and victim model are trained on different datasets. We denote this challenging setting as **g**eneralized

Table 1: The information that the attack methods can access.

Attack Setting	Information of the target model that can be accessed
White-Box	All information (network architecture, network weight, gradient, score, prediction, input-resolution, output-dimension, output-classes, *etc.*)
Query-based Black-Box	Limited information (prediction, score, input-resolution, image classes)
Query-free Black-Box	Limited information (input-resolution, image classes)
Generalized Transferable Attack	Non information (/)

transferable **a**ttack (**GTA**) setting because the attacker needs to be generalizable to attack **images from unknown datasets** and **any models** predicting these images. Table 1 illustrates the difference between GTA and the previous attack settings.

Under the GTA setting, we aim at investigating whether DNNs are still vulnerable and whether there exists new attacks in this setting to break DNN models. Although none of the previous transfer attack paper considers attacking across datasets, with some careful modifications on the attack loss and rescaling techniques, it is possible to extend existing transfer attacks to this new setting (we will discuss these modifications in Section 4.1). Unfortunately, as will be seen in the experimental results, even with these modifications the existing transfer attacks suffer from very poor attack success rates due to the mis-match of label set and input size between source and target models. To tackle these challenges, we propose a novel method called Image Classification Eraser (ICE) which builds a generalized attacker by a meta-learning framework (Finn et al., 2017; Mishra et al., 2018; Qin et al., 2021b; Liu et al., 2019b). ICE requires no assumption on the dataset of the victim model, and furthermore, it allows multiple white-box surrogate models trained on different datasets, with various label sets and input sizes. Extensive experiments on Cifar-10 (Krizhevsky et al., 2009), Cifar-100, and TieredImageNet (Ren et al., 2018a) demonstrate that the proposed ICE outperforms the modified transfer attack methods on the GTA problem. In particular, given the source dataset Cifar-10 and the source models ResNet-18 and MobileNet-V1 trained on Cifar-10, ICE improves the average attack success rate on Cifar-100 images by about 17.0%, compared with existing transfer attack methods.

2 Background

Existing adversarial attack methods (Ganeshan et al., 2019; Croce & Hein, 2020a; Wu et al., 2020b; Li et al., 2020b; Kaidi et al., 2019; Maksym et al., 2020; Xiao et al., 2021; Zhang et al., 2021) can mainly be categorized into white-box, query-based black-box, and query-free black-box attacks. Among the three kinds of attacks, white-box attack (Kurakin et al., 2016) is the most effective one because all information of the target model can be leveraged to generate adversarial examples. Query-based black-box attack assumes that some information of the target model is hidden to users and the users can only query the target model and access the hard-label or soft-label predictions. Researchers have proposed many query-based black-box attack methods (Chen et al., 2017; Cheng et al., 2020; Huang & Zhang, 2020; Gao et al., 2020) and have shown that adversarial examples can still be effectively generated only based on predictions. The most recently developed query-based methods mainly focus on improving the querying efficiency and reducing the query counts (Cheng et al., 2019; Li et al., 2020a; Du et al., 2020; Wang et al., 2020; Yuan et al., 2021). The query-free black-box attack further assumes that the target model's prediction is also hidden to users. In this challenging situation, researchers usually generate adversarial examples by attacking surrogate models (Dong et al., 2019; Xie et al., 2019). Then, by leveraging the transferability of adversarial examples (Papernot et al., 2016; Tramèr et al., 2017; Nathan et al., 2020), we can directly use the adversarial examples to attack the target model without querying (Huang et al., 2019; Zhou et al., 2018; Lu et al., 2020). However, exising works commonly assume the surrogate model and victim model are trained on the same dataset, which indicates that the attacker implicitly knows the dataset, input size, and label set used in the victim model.

The level of information that the attacker can leverage reduces from white-box to query-free black-box attacks. However, existing methods still need to know some information of the target model, which is summarized in Table 1. In this paper, instead of following the above three adversarial attack directions, we consider a novel and more challenging problem called generalized transferable attack (GTA). GTA can be described as the following attack scene. We have the resources of: 1) some source datasets. 2) some source models trained on the source datasets. With these resources, we can obtain an attacker. Then, given a randomly intercepted image, we are required to directly leverage the attacker to disturb the image so that any unknown target models that predict this image will make wrong predictions for the disturbed image. Note that the image is randomly intercepted and it is normal that the source datasets do not contain the image category of the randomly intercepted image. Further, both the resolution of the image and that of the target model cannot be known in advance.

3 METHODOLOGY

In GTA, we assume the source models can be trained on several datasets that are different from the victim model. Suppose we have m source image classification datasets denoted as $\mathcal{D}_1$, $\mathcal{D}_2$, ..., $\mathcal{D}_m$. Different datasets have different label sets, with potentially different label sizes and image shapes. For each source dataset $\mathcal{D}_k$, we have N_k trained models denoted as $\mathbf{M}_{\mathcal{D}_k}$ = { $\mathbf{M}^1_{\mathcal{D}_k}$, $\mathbf{M}^2_{\mathcal{D}_k}$, ..., $\mathbf{M}^{N_k}_{\mathcal{D}_k}$ }. With these resources, we can build a model $\mathcal{A}(\mathbf{M}_{D_1}, \mathbf{M}_{D_2}, ..., \mathbf{M}_{D_k})$, which may be a simple ensemble of all source models or a new model obtained by using the sources. Then, given any encountered image x in the inference time, we generate an adversarial image via the formulation $\hat{x} = f(\mathcal{A}(\mathbf{M}_{D_1}, \mathbf{M}_{D_2}, ..., \mathbf{M}_{D_k}), x)$, where f is a gradient-based attack to obtain $\hat{x}$ by attacking $\mathcal{A}$. The goal of GTA is that any unknown model $\mathbb{M}$ that takes x as input will be fooled by $\hat{x}$, which can be formulated as $\mathbb{M}(\hat{x}) \neq \mathbb{M}(x)$.

A traditional way in transfer attack is to build the model $\mathcal{A}$ as the ensemble of all surrogate models (Dong et al., 2018; Wu et al., 2020a). However, since source models can have different input shapes and label spaces in the GTA setting, it is nontrivial to ensemble them into a single model. Furthermore, a naive ensemble may not optimize the performance for generalized transfer attack. Therefore, we propose a novel method called image classification eraser (ICE) to obtain a single model $\mathcal{A}$ that optimizes the performance of GTA attack using meta-learning (Finn et al., 2017; Qin et al., 2020; Javed & White, 2019). This model can be understood as a universal surrogate model and to distinguish it from the naively assembled model, it will be denoted as $\mathcal{U}_\theta$ in the rest of this paper, where θ is the model parameter. It is expected that by confusing the model $\mathcal{U}_\theta$ with function f (PGD in our work), we can obtain an adversarial example for the image x to fool an unknown target model. The design of our method aims to address the following issues in generalized transfer attack:

1) We cannot predict the category of a randomly encountered image x in advance, so no ground-truth information can be leveraged. This means the commonly used cross-entropy loss which needs the ground-truth label cannot be directly used in GTA. We therefore use entropy instead of cross-entropy for the attack. Eq. 1 shows the formulations of cross-entropy $\mathrm{CE}(d, y)$ and entropy $\mathcal{L}(d)$, where d is a vector that denotes the prediction distribution and y is the one-hot ground-truth label.

$$\begin{cases} d & = \mathrm{Softmax}(\mathrm{logit}) \\ \mathcal{L}(d) & = -d^T \cdot \log(d) \\ \mathrm{CE}(d, y) & = -y^T \cdot \log(d) \end{cases} \tag{1}$$

Entropy of any distribution denotes the degree of disorder, randomness, or uncertainty of the distribution. Therefore, by maximizing the entropy, we can obtain a perturbation that makes the input image hard to be classified without using the ground-truth label. Furthermore, the entropy loss also enables us to flexibly set the output dimension of model $\mathcal{U}_\theta$ without knowing the number of categories of any dataset. In the experiments, we set the output dimension of $\mathcal{U}_\theta$ to 1000 by default and investigate the impact of the output dimension on its performance in an ablation study.

2) Random encountered images may have diverse shapes. For example, the image-shapes from different datasets (*e.g.*, Cifar-10 and ImageNet) differ from each other. Therefore, the model $\mathcal{U}_\theta$ must be capable of directly handling images with different shapes without resizing images. To guarantee this point, we set the network architecture of model $\mathcal{U}_\theta$ to a fully convolutional network without flattened and fully-connected layers. All down-sampling operations are implemented by max-pooling and average-pooling. We will show details of the network architecture in Section A.1.

3.1 IMAGE CLASSIFICATION ERASER

To optimize the model $\mathcal{U}_\theta$, we use all source models from source datasets to simulate unknown target models and develop a novel bi-level training framework (Finn et al., 2017; Liu et al., 2019b; Ren et al., 2018b; Liu et al., 2019a). Each bi-level training iteration contains an inner-loop and an outer-loop optimization. In the inner-loop, by confusing model $\mathcal{U}_\theta$ with gradient ascent, we generate adversarial examples for any source images and feed them into the source models to simulate the GTA process.

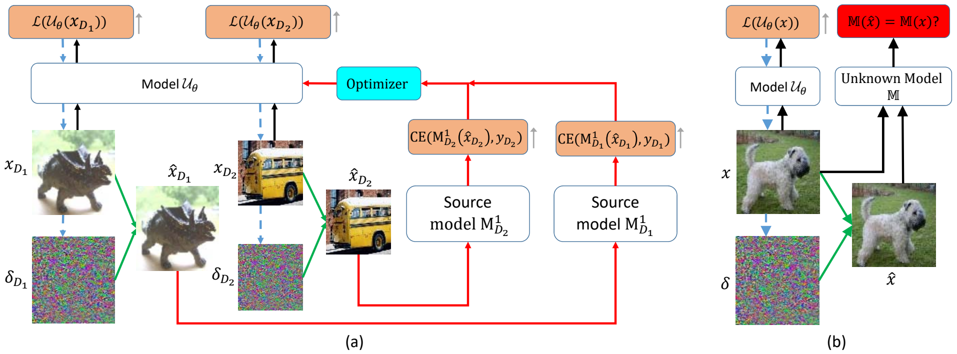
$\mathcal{L}(\mathcal{U}_\theta(x_{D_1}))$
$\mathcal{L}(\mathcal{U}_\theta(x_{D_2}))$
Model $\mathcal{U}_\theta$
Optimizer
$\text{CE}(\text{M}^1_{D_2}(\hat{x}_{D_2}), y_{D_2})$
$\text{CE}(\text{M}^1_{D_1}(\hat{x}_{D_1}), y_{D_1})$
x_{D_1}
$\hat{x}_{D_1}$
x_{D_2}
$\hat{x}_{D_2}$
δ_{D_1}
δ_{D_2}
Source model $\text{M}^1_{D_2}$
Source model $\text{M}^1_{D_1}$
(a)
$\mathcal{L}(\mathcal{U}_\theta(x))$
$\mathbb{M}(\hat{x}) = \mathbb{M}(x)$?
Model $\mathcal{U}_\theta$
Unknown Model $\mathbb{M}$
x
δ
$\hat{x}$
(b)

Figure 1: (a) The training framework of the proposed ICE. Different images sampled from different source datasets are used to mimic randomly encountered images and are simultaneously fed into $\mathcal{U}_\theta$. By maximizing the prediction entropy of model $\mathcal{U}_\theta$ for the input images, we obtain adversarial images. To evaluate how confusing the generated adversarial images are, we feed them into the source models and measure the cross-entropy loss. Finally, we optimize $\mathcal{U}_\theta$ by maximizing the source models' cross-entropy losses. (b) The testing pipeline of the proposed ICE.

In the outer-loop, we evaluate how the source models are being confused by the adversarial examples generated in the inner-loop, and optimize the model $\mathcal{U}_\theta$ by maximizing the cross-entropy loss of the source models. Details of the inner-loop and outer-loop can be described as the follows.

Inner-Loop. Given any source dataset D_k and any image $x_{\mathcal{D}_k} \in \mathcal{D}_k$, we simulate it as a randomly encountered image and conduct a GTA process to $x_{\mathcal{D}_k}$. Specifically, we firstly feed $x_{\mathcal{D}_k}$ into $\mathcal{U}_\theta$ and obtain the prediction $\mathcal{U}_\theta(x_{\mathcal{D}_k})$, and then we generate adversarial example $\hat{x}_{\mathcal{D}_k}$ by maximizing the entropy of $\mathcal{U}_\theta(x_{\mathcal{D}_k})$. To enable the gradient back-propagating in the bi-level optimization framework, we follow the transferable attack method MTA (Qin et al., 2021a) to maxmize the model $\mathcal{U}_\theta$'s prediction entropy via one-step Customized PGD (Customized FGSM). The reason why we use Customized FGSM instead of multi-step Customized PGD will be described in Section 4.5.4. Then, we can formulate $\hat{x}_{\mathcal{D}_k}$ as

$$\begin{cases} g(\theta) = \nabla_{x_{\mathcal{D}_k}} \mathcal{L}(\mathcal{U}_\theta(x_{\mathcal{D}_k})) \\ \hat{x}_{\mathcal{D}_k} = \text{Clip}\Big(x_{\mathcal{D}_k} + \epsilon_c \cdot \big(\gamma_1 \cdot \frac{g(\theta)}{\text{sum}(\text{abs}(g(\theta)))} + \gamma_2 \cdot \frac{2}{\pi} \cdot \arctan(\frac{g(\theta)}{\text{mean}(\text{abs}(g(\theta)))}) + \text{sign}(g(\theta))\big)\Big) \end{cases} \tag{2}$$

where both γ_1 and γ_2 are set to 0.01 by default. ϵ_c determines the perturbation scale. $\mathcal{L}(\mathcal{U}_\theta(x_{\mathcal{D}_k}))$ is the entropy of $\mathcal{U}_\theta(x_{\mathcal{D}_k})$, and $g(\theta)$ is the gradient of the entropy *w.r.t* $x_{\mathcal{D}_k}$ based on the current parameter θ. Clip is the function that clips each pixel value of the image into the range of [0, 255].

Outer-Loop. We evaluate how the perturbed image $\hat{x}_{\mathcal{D}_k}$ fools each simulated unknown target model $\mathbf{M}^j_{\mathcal{D}_k} \in \mathbf{M}_{\mathcal{D}_k}$ by calculating the adversarial loss

$$\boldsymbol{l}^j_{\mathcal{D}_k} = \text{CE}(\mathbf{M}^j_{\mathcal{D}_k}(\hat{x}_{\mathcal{D}_k}), y_{\mathcal{D}_k}), \tag{3}$$

where $j \in [1, N_k]$; $y_{\mathcal{D}_k}$ is the groundtruth of $x_{\mathcal{D}_k}$; CE is the cross-entropy function. $\mathbf{M}^j_{\mathcal{D}_k}(\hat{x}_{\mathcal{D}_k})$ is the target model's prediction for $\hat{x}_{\mathcal{D}_k}$. A larger adversarial loss $\boldsymbol{l}^j_{\mathcal{D}_k}$ will indicate a higher possibility that the simulated unknown target model $\mathbf{M}^j_{\mathcal{D}_k}$ is fooled by the perturbed image $\hat{x}_{\mathcal{D}_k}$. Note that the groundtruths of all source images are accessible when we training the model $\mathcal{U}_\theta$, so we use cross-entropy instead of entropy used in inner-loop to calculate the loss in outer-loop. The ablation study in Section 4.5.5 validates the necessity of the cross-entropy for the outer-loop.

Algorithm 1: Training of the Image Classification Eraser

```
input: Source datasets $\mathbb{D} = \{\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_m\}$, Source models $\mathbf{M}_{\mathcal{D}_k} = \{\mathbf{M}^1_{\mathcal{D}_k}, \mathbf{M}^2_{\mathcal{D}_k}, ..., \mathbf{M}^{N_k}_{\mathcal{D}_k}\}$ for each dataset $\mathcal{D}_k$.
output: Optimized weight $\theta$.
while not done do
    for each $\mathcal{D}_k \in \mathbb{D}$ do
        Sample a mini data batch $(X_{\mathcal{D}_k}, Y_{\mathcal{D}_k}) \in \mathcal{D}_k$
        Obtain adversarial examples $\hat{X}_{\mathcal{D}_k}$ via Eq. 2
        for each $\mathbf{M}^j_{\mathcal{D}_k} \in \mathbf{M}_{\mathcal{D}_k}$ do
            Obtain adversarial loss $\mathbf{L}^j_{\mathcal{D}_k}$ for $\hat{X}_{\mathcal{D}_k}$ via Eq. 3.
        end for
    end for
    $\theta = \theta + \alpha \cdot \nabla_\theta \big(\frac{1}{m} \sum_{k=1}^{m} (\frac{1}{N_k} \sum_{j=1}^{N_k} \mathbf{L}^j_{\mathcal{D}_k})\big)$
end while
return $\theta$
```

To ensure that the classification information of each image $x_{D_k} \in \mathcal{D}_k$ can be erased by attacking model $\mathcal{U}_\theta$ and the perturbed image $\hat{x}_{D_k}$ is confusing for the simulated unknown target model $\mathbf{M}^j_{\mathcal{D}_k}$ to predict, we optimize the model $\mathcal{U}_\theta$ by maximizing the adversarial loss $\boldsymbol{l}^j_{\mathcal{D}_k}$ by the following SGD update:

$$\theta = \theta + \alpha \cdot \nabla_\theta \boldsymbol{l}^j_{\mathcal{D}_k}, \tag{4}$$

where α is the learning rate. $\boldsymbol{l}^j_{\mathcal{D}_k}$ is differentiable *w.r.t* θ because $\boldsymbol{l}^j_{\mathcal{D}_k}$ depends on $\hat{x}_{\mathcal{D}_k}$ and $\hat{x}_{\mathcal{D}_k}$ depends on θ. In our experiment, we optimize the model $\mathcal{U}_\theta$ by simultaneously maximizing the adversarial losses on all source models from all source datasets in each iteration, which is summarized in Algorithm 1. This procedure will enforce $\mathcal{U}_\theta$ having the property that the adversarial examples constructed by attacking it are more transferable to images from different datasets.

3.2 Inference procedure and evaluation

Given any clean image x that will be fed into an unknown target model $\mathbb{M}$, we evaluate the proposed ICE with the following steps. 1) Directly feed the image x into model $\mathcal{U}_\theta$ and generate the adversarial example $x^{(T)}$ by maximizing the entropy for T gradient ascent steps. The i-th step is formulated as

$$\begin{cases} \delta^{(i-1)} = \text{sign}\big(\nabla_{x^{(i-1)}} \mathcal{L}(\mathcal{U}_\theta(x^{(i-1)}))\big), \\ x^{(i)} = \text{clip}(x^{(i-1)} + \frac{\epsilon}{T} \cdot \delta^{(i-1)}), \end{cases} \tag{5}$$

where $x^{(0)} = x$ and $\delta^{(i-1)}$ is the perturbation generated in the i-th step. ϵ/T is the L_∞ perturbation scale in each step. 2) Generate the adversarial example $\hat{x}$ by the formulation $\hat{x} = \text{clip}(x + \epsilon \cdot \text{sign}(x^{(T)} - x))$. We call this step 'Sign-Projection' (SP). The reason why we use SP is that it enlarges the average distortion of each pixel without amplifying the L_∞ norm of the perturbation, which improves the GTA success rate. Ablation study in Section 4.5.1 will show that SP improves both ICE and the baselines introduced in Section 4.1. 3) Feed the adversarial example $\hat{x}$ and the clean image x into the unknown target model and get its predictions $\mathbb{M}(\hat{x})$ and $\mathbb{M}(x)$. 4) The GTA process is successful if $\mathbb{M}(\hat{x}) \neq \mathbb{M}(x)$. The evaluation pipeline is also illustrated in Figure 1b, where the green arrows indicate the second step.

4 Experimental Results

In this section, we conduct several experiments to evaluate the proposed method for conducting generalized transferable attacks. Three datasets Cifar-10 (Krizhevsky et al., 2009), Cifar-100, and TieredImageNet (Ren et al., 2018a) are used to build the testing scenes of GTA. Both Cifar-10 and Cifar-100 contain 60,000 images with the resolution of 32×32. TieredImageNet is a subset sampled from ImageNet (Deng et al., 2009). The default image resolution in TieredImagenet is 84x84. We split TieredImageNet into two datasets. The training set of TieredImageNet is treated as one dataset and is denoted as Tiered$_{T84}$. The validation and the testing sets with all images resized into 56×56 resolution are treated as another dataset, denoted as Tiered$_{V56}$. For either Tiered$_{T84}$ or Tiered$_{V56}$, we use the first 1200 images of each category to compose the training set and use the last 100 images to compose the testing set. Overall, we have **four** datasets **Cifar-10**, **Cifar-100**, **Tiered**$_{T84}$, and **Tiered**$_{V56}$ that will be used in our experiments. More details of Tiered$_{T84}$, and Tiered$_{V56}$ will be shown in Section A.2.

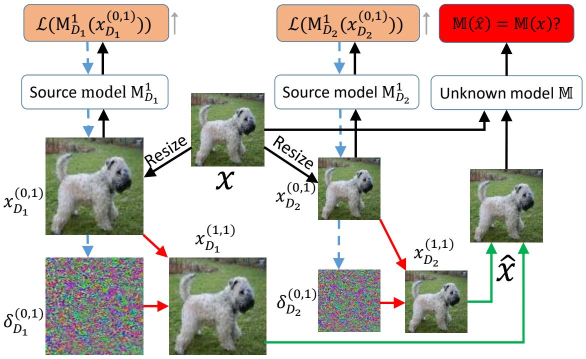

Figure 2: Inference pipeline of PGD-based baselines.

4.1 How to use transfer attack baselines for GTA?

GTA is a novel adversarial attack problem and few existing methods can be directly used as baselines. Considering that transfer attack is the most similar problem to GTA, we deploy several transferable adversarial attack methods including MI (Dong et al., 2018), DI (Xie et al., 2019), TI-DIM (Dong et al., 2019), SGM (Wu et al., 2020a), AEG (Bose et al., 2020), IR (Wang et al., 2021), and MTA (Qin et al., 2021a) on GTA as baselines for the proposed ICE. Except for AEG and MTA, all the other baselines are implemented on the GTA problem with the inference pipeline described below. The detailed implementations of AEG and MTA will be introduced in Section A.3.

1) Since the source models trained on different source datasets commonly have different input shapes, we firstly resize the testing image x to the input shapes of all source models and then feed the resized images to the source models, respectively. Then the inference of x on one source model $\mathbf{M}_{D_k}^{j}$ can be formulated as $y_{D_k}^{(0,j)} = \mathbf{M}_{D_k}^{j}(x_{D_k}^{(0,j)})$, where $x_{D_k}^{(0,j)} = \text{resize}(x, \text{resolution}(\mathbf{M}_{D_k}^{j}))$. 2) Because we cannot access the category of the image x in advance, no ground-truth label can be leveraged to perturb the resized image. Therefore, for each source model, we generate adversarial perturbation by maximizing the entropy (as used in our ICE) for T gradient ascent steps. The i-th step can be formulated as

$$\begin{cases} y_{D_k}^{(i-1,j)} = \mathbf{M}_{D_k}^{j}(x_{D_k}^{(i-1,j)}), \\ \delta_{D_k}^{(i-1,j)} = \text{sign}\big(\nabla_{x_{D_k}^{(i-1,j)}} \mathcal{L}(y_{D_k}^{(i-1,j)})\big), \\ x_{D_k}^{(i,j)} = \text{clip}(x_{D_k}^{(i-1,j)} + \frac{\epsilon}{T} \cdot \delta_{D_k}^{(i-1,j)}), \end{cases} \tag{6}$$

where $y_{D_k}^{(i-1,j)}$ is the source model's output and $x_{D_k}^{(i,j)}$ and $\delta_{D_k}^{(i-1,j)}$ are the adversarial example and the perturbation generated in the i-th step, respectively. 3) We resize all the adversarial examples to the original shape of the image x and average fuse all adversarial examples to one image x_{adv} following the formulation $x_{adv} = \frac{1}{m} \cdot \sum_{k=1}^{m} \cdot \big(\frac{1}{N_k} \cdot \sum_{j=1}^{N_k} \text{resize}(x_{D_k}^{(T,j)}, \text{resolution}(x))\big)$. 4) We generate adversarial example $\hat{x}$ by the formulation $\hat{x} = \text{clip}(x + \epsilon \cdot \text{sign}(x_{adv} - x))$, which is the SP step defined in Section 3.2. 5) We feed the adversarial example $\hat{x}$ and the clean image x into the unknown target model $\mathbb{M}$ and get its predictions. 6) The GTA process is successful if $\mathbb{M}(\hat{x}) \neq \mathbb{M}(x)$. The evaluation of the PGD-based baselines is also illustrated in Figure 2, where the green arrows denote the third and fourth steps.

4.2 Experimental settings

1) **Source and Target models**. On each dataset, we train several models including ResNet-18, -34 (He et al., 2016), SeResNet-26 (Hu et al., 2018), VGG-16 (Simonyan & Zisserman, 2015), MobileNet-V1 (Howard et al., 2017), MobileNet-V3 (Howard et al., 2019), and DenseNet-26 (Huang et al., 2017). The training details and the architectures of these models will be introduced in Section A.4. These models will be used as the source and target models in the following GTA testing experiments.

2) **Hyper-parameters**. In the training phase of ICE, we train $\mathcal{U}_\theta$ for 50,000 iterations with batch size of 64 for each source dataset. Learning rate α is set to 0.01. ϵ_c is set to 3000 and is periodically decayed by 0.9× for every 3000 iterations. In the inference phase of ICE and all the baselines, we set ϵ to 15 and set T to 10 for ICE and other PGD-based baselines. In Section A.5.1, We will show the experimental results with $\epsilon = 8$, where ICE still outperforms the baselines with significant margins.

3) **Evaluation Metric**. We use the attack success rate to denote the GTA performance. Because attacking the images that are wrongly classified by the target model is meaningless, we only attack the images that are correctly classified by the target model.

4.3 Generalized transferable attack to Cifar-100

In this subsection, we perform GTA experiments on Cifar-100, which means we try to perturb Cifar-100 images to fool the target victim models. The models MobileNet-V3, VGG-16, ResNet-18, ResNet-34, SeResNet-26, and DenseNet-26 trained on Cifar-100 are used as victims to calculate the GTA success rate. All experimental results are reported in Table 2.

In the first experiment (the first row of Table 2), we use ResNet-18 and Cifar-10 as the source model and the source dataset, respectively, and conduct GTA on the testing images from Cifar-100. It is observed that among all baselines, FGSM performs the best. A possible underlying reason is that adversarial perturbations generated by multi-step gradient ascent tends to overfit the source model and source dataset. It can also be seen that the proposed ICE outperforms existing methods on the GTA problem. For instance, compared with FGSM, the average attack success rate on the six target models is improved by about 11.5%.

In the second experiment (the second row of Table 2), we consider the case where there are two source models – ResNet-18 and MobileNet-V1 trained on Cifar-10. Experimental results indicate that most of the baselines cannot leverage the additional source model MobileNet-V1 to boost their performances. In contrast, the proposed ICE can efficiently make use of the additional source model to improve its performances. For instance, by adding the MobileNet-V1 source model, the average attack success rate across 6 models of FGSM is decreased by about 0.7% while the success rate of ICE is improved by about 11.6%.

Table 2: GTA success rates on Cifar-100.

Resource	Method	MobileNet-V3	VGG-16	ResNet-18	ResNet-34	SeResNet-26	DenseNet-26
Cifar-10 (ResNet-18)	FGSM	47.7%	64.2%	59.1%	57.6%	59.1%	73.1%
	PGD	37.3%	50.9%	43.2%	44.5%	45.7%	61.5%
	DI	39.3%	54.9%	47.2%	46.1%	49.5%	64.7%
	MI	45.4%	62.3%	56.3%	56.6%	57.5%	72.1%
	TI-DIM	51.0%	45.6%	48.0%	47.7%	47.5%	55.3%
	IR	48.3%	62.1%	60.5%	58.8%	59.1%	73.7%
	AEG	51.3%	61.7%	61.2%	59.5%	58.9%	66.4%
	MTA	42.3%	49.5%	45.0%	45.1%	44.2%	60.0%
	ICE	**54.9%**	**67.3%**	**71.5%**	**64.0%**	**61.2%**	**83.5%**
Cifar-10 (ResNet-18 +MobileNet-V1)	FGSM	49.2%	63.3%	57.9%	56.4%	58.9%	72.5%
	PGD	39.6%	52.8%	45.1%	44.6%	47.9%	62.9%
	DI	42.1%	55.2%	46.2%	45.5%	49.8%	65.0%
	MI	47.4%	62.7%	57.1%	56.1%	58.3%	72.1%
	TI-DIM	51.5%	43.6%	45.5%	46.0%	46.1%	52.6%
	IR	52.6%	63.8%	60.3%	58.8%	59.0%	74.9%
	AEG	55.8%	65.3%	65.0%	62.8%	63.9%	71.2%
	MTA	45.2%	54.1%	51.1%	49.8%	49.2%	62.6%
	ICE	**55.2%**	**79.6%**	**79.9%**	**77.5%**	**72.7%**	**84.3%**
Cifar-10 + Tiered$_{T84}$ (ResNet-18)	FGSM	33.0%	47.3%	38.6%	37.2%	40.5%	58.7%
	PGD	39.0%	52.2%	41.7%	41.3%	46.8%	64.9%
	DI	39.1%	52.3%	41.7%	41.4%	47.3%	64.2%
	MI	42.7%	57.0%	48.2%	47.7%	52.5%	68.1%
	TI-DIM	55.1%	48.2%	50.5%	50.6%	49.8%	58.1%
	IR	44.0%	58.1%	51.9%	50.4%	52.9%	69.5%
	AEG	50.9%	64.6%	58.2%	55.6%	58.5%	70.3%
	MTA	43.6%	56.7%	47.5%	47.7%	51.3%	68.0%
	ICE	**57.0%**	**76.2%**	**77.5%**	**76.8%**	**69.3%**	**83.3%**
Cifar-10 + Tiered$_{V56}$ (ResNet-18)	FGSM	33.7%	48.5%	38.9%	38.7%	41.6%	58.2%
	PGD	40.1%	54.9%	44.3%	43.5%	48.9%	65.7%
	DI	40.5%	55.0%	44.7%	43.5%	49.0%	65.9%
	MI	45.5%	60.3%	50.2%	48.9%	53.8%	70.6%
	TI-DIM	51.4%	48.0%	49.4%	49.5%	48.6%	58.9%
	IR	44.7%	59.2%	54.0%	53.5%	53.9%	70.1%
	AEG	51.5%	62.3%	58.9%	56.4%	57.1%	68.3%
	MTA	43.3%	60.8%	50.0%	50.3%	53.5%	70.2%
	ICE	**52.7%**	**76.9%**	**79.8%**	**78.0%**	**69.5%**	**84.9%**
Cifar-10 + Tiered$_{T84}$ + Tiered$_{V56}$ (ResNet-18)	FGSM	45.3%	59.6%	49.9%	48.2%	53.4%	70.0%
	PGD	40.6%	54.7%	42.8%	42.1%	45.6%	65.8%
	DI	40.6%	54.9%	42.9%	42.2%	48.6%	66.0%
	MI	45.0%	58.9%	48.1%	47.0%	54.0%	69.2%
	TI-DIM	54.3%	48.7%	50.1%	51.3%	49.8%	58.7%
	IR	48.9%	57.5%	51.2%	50.8%	56.2%	70.1%
	AEG	47.7%	63.8%	56.0%	53.1%	57.5%	71.7%
	MTA	40.6%	61.3%	49.2%	50.1%	53.6%	69.7%
	ICE	**56.3%**	**83.2%**	**90.1%**	**87.3%**	**80.1%**	**92.4%**

In the third experiment, we use two ResNet-18 models trained on Cifar-10 and Tiered$_{T84}$ respectively as the source models, and use Cifar-10 and Tiered$_{T84}$ as the source datasets. It is interesting that the baselines' performances in this experiment are commonly worse than their performances in the first experiment. The possible reason for this result is that the resolution of the images from Tiered$_{T84}$ is 84×84, which differs greatly from the resolution of Cifar-100. As a comparison, ICE's performances in this experiment are much better than its performances in the first experiment, which indicates that ICE can efficiently make use of all the resources to improve the performance in spite of the difference among the source datasets.

In the fourth experiment, we use two ResNet-18 models respectively trained on Cifar-10 and Tiered$_{T56}$ as the source models. It is observed that most of the baselines' performances in this experiment are slightly better than their performances in the third experiment. For instance, compared with the performances in the third experiment, the performance of FGSM in this experiment is improved by about 1.7%. The possible reason for this result is that compared with the resolution of Tiered$_{T84}$, the resolution of Tiered$_{V56}$ is more closer to the resolution of Cifar-100. In this experiment, the proposed ICE still outperforms all baselines with clear margins.

In the fifth experiment, we use three ResNet-18 models respectively trained on Cifar-10, Tiered$_{T84}$ and Tiered$_{V56}$ as the source models. It is clear that ICE's performances can be further improved by using more source datasets, while the baselines cannot efficiently utilize the additional source models to achieve better performances. AEG performs the best among all baselines with the average attack success rate of 58.3%. Compared with AEG, ICE promotes the average attack success rate

Table 3: The GTA success rates on Cifar-10, Tiered$_{T84}$, and Tiered$_{V56}$.

Resource	Method	MobileNet-V3	VGG-16	ResNet-18	ResNet-34	SeResNet-26	DenseNet-26
-Cifar-10 (ResNet-18)	FGSM	21.3%	28.3%	33.0%	30.2%	26.2%	47.9%
	PGD	17.7%	24.7%	28.4%	25.7%	22.0%	43.2%
	DI	17.6%	24.8%	28.4%	25.7%	22.0%	43.3%
	MI	20.6%	28.7%	33.2%	30.2%	26.1%	47.8%
	TI-DIM	15.5%	18.5%	22.2%	19.5%	17.6%	25.2%
	IR	19.8%	28.3%	32.3%	29.7%	26.5%	47.9%
	AEG	24.4%	38.5%	41.2%	40.9%	31.3%	52.1%
	MTA	22.5%	39.1%	41.2%	39.6%	32.6%	53.0%
	ICE	**36.3%**	**45.5%**	**61.0%**	**53.6%**	**48.5%**	**65.5%**
-Tiered$_{T84}$ (ResNet-18)	FGSM	56.5%	62.1%	83.5%	77.2%	67.3%	88.7%
	PGD	50.1%	53.3%	75.5%	68.8%	57.5%	83.0%
	DI	50.5%	53.3%	75.5%	69.5%	57.5%	83.6%
	MI	**58.2%**	61.8%	82.6%	77.1%	66.4%	88.3%
	TI-DIM	56.8%	52.6%	77.3%	71.1%	57.5%	80.9%
	IR	52.5%	56.8%	79.3%	72.7%	62.9%	87.5%
	AEG	47.8%	45.6%	56.0%	55.7%	51.0%	64.5%
	MTA	42.2%	43.8%	53.2%	52.5%	50.6%	60.5%
	ICE	52.3%	**73.0%**	**90.8%**	**89.6%**	**73.8%**	**93.5%**
-Tiered$_{V56}$ (ResNet-18)	FGSM	65.7%	69.8%	78.1%	75.0%	75.1%	86.9%
	PGD	59.5%	65.8%	74.1%	70.9%	71.1%	85.8%
	DI	59.3%	65.8%	74.3%	71.1%	70.9%	85.6%
	MI	64.6%	68.0%	76.2%	73.9%	73.7%	86.5%
	TI-DIM	63.2%	60.0%	71.2%	69.3%	65.8%	76.9%
	IR	59.7%	67.1%	74.3%	73.2%	69.0%	83.5%
	AEG	57.0%	67.2%	72.7%	70.3%	72.0%	78.6%
	MTA	48.0%	63.9%	66.7%	64.2%	65.9%	72.5%
	ICE	**72.0%**	**83.5%**	**91.7%**	**88.2%**	**85.3%**	**93.3%**

by about 39.9%, which is a bigger margin than the margin in the first experiment. This experiment further indicates that ICE is more effective than baselines to leverage all the resource to solve the GTA problem.

4.4 Generalized transferable attack to Cifar-10, Tiered$_{T84}$, and Tiered$_{V56}$

We have performed GTA on Cifar-100 in the previous subsection. Now we show ICE still outperforms baselines when using other datasets as target images. Table 3 reports the experimental results. There are four datasets in total (Cifar-10, Cifar-100, Tiered$_{T84}$, and Tiered$_{V56}$), and each row (denoted as -target) shows the experiment when conducting GTA on the target dataset by using ResNet-18 trained on the other three datasets as source models. For example, the '-Cifar10' row denotes the experiment that utilizing the datasets Cifar-100, Tiered$_{T84}$, and Tiered$_{V56}$ and the three respectively trained ResNet-18 models to conduct GTA to the images from Cifar-10. Table 3 does not show the '-Cifar100' row because the corresponding results have been shown in the last row of Table 2. It is clear that given three datasets and the models trained on the three datasets, ICE performs the best to attack unknown images from other datasets.

4.5 Ablation Study

Here we conduct several ablation studies to verify the effect of each setting or component in our work. Note that the source model and source dataset used in all ablation experiments are ResNet-18 and Cifar-10, and the target dataset is Cifar-100.

4.5.1 The effect of SP

When evaluating ICE and baselines on the GTA problem, we use the trick SP (the third step in Section 3.2) to improve their performances. Here we validate the effectiveness of SP by removing it, which means we directly use the adversarial example $x^{(T)}$ generated in the second step in Section 3.2 and x_{adv} generated in the third step in Section 4.1 to attack the target models. Table 4 reports the experimental results. It is clear that without SP, the performances of ICE and baselines are greatly damaged. The reason is that SP enlarges the average perturbation scale of each pixel, which possibly is an important factor for GTA. More analyses about SP will be shown in Section A.7.

4.5.2 Using pseudo label with cross-entropy loss?

Section 4.1 introduces that for ICE and all baseline methods, we generate adversarial examples by maximizing the entropy. Here we use pseudo label with cross-entropy loss to generate adversarial examples. Then the perturbation generated in each gradient ascent step can be reformulated as $\delta = \text{sign}\big(\nabla_{x^{(i-1)}}\text{CE}(y, \text{argmax}(y))\big)$, where y is the softmax prediction and $\text{argmax}(y)$ is used as

Table 4: Ablation GTA experiments on Cifar-100.

Setting	Method	MobileNet-V3	VGG-16	ResNet-18	ResNet-34	SeResNet-26	DenseNet-26
w/o SP	PGD	7.0%	13.9%	12.3%	11.7%	12.6%	19.1%
	DI	6.9%	13.0%	11.7%	10.7%	12.1%	18.5%
	MI	**33.8%**	**50.9%**	**45.3%**	**45.0%**	**45.0%**	**60.7%**
	TI-DIM	10.0%	8.0%	8.5%	8.3%	8.9%	12.1%
	ICE	21.9%	23.6%	23.3%	20.0%	22.1%	35.1%
Pseudo	PGD	38.9%	53.0%	45.2%	44.9%	48.1%	63.3%
	DI	40.9%	53.5%	46.0%	45.1%	50.1%	64.2%
	MI	**47.3%**	**62.4%**	**57.3%**	**56.2%**	**57.5%**	**72.2%**
	TI-DIM	**47.3%**	41.5%	42.6%	43.6%	43.9%	51.3%
	ICE	39.1%	55.8%	53.6%	48.9%	46.3%	70.8%
Default	PGD	37.3%	50.9%	43.2%	44.5%	45.7%	61.5%
	DI	39.3%	54.9%	47.2%	46.1%	49.5%	64.7%
	MI	45.4%	62.3%	56.3%	56.6%	57.5%	72.1%
	TI-DIM	51.0%	45.6%	48.0%	47.7%	47.5%	55.3%
	ICE$_{en}$	39.3%	49.0%	46.5%	42.6%	52.9%	66.3%
	ICE	**54.9%**	**67.3%**	**71.5%**	**64.0%**	**61.2%**	**83.5%**

pseudo label. $x^{(i-1)}$ is the adversarial image we obtained after $i-1$ gradient ascent steps. The corresponding experimental results on Cifar-100 are shown in Table 4. It can be seen that for most of the baselines, the pseudo label together with cross-entropy loss perform almost consistent with entropy in GTA. For the proposed ICE, the cross-entropy loss damages the performance, which is probably caused by the fact that ICE is trained with entropy but not cross-entropy.

4.5.3 Output dimension of ICE

We set the output dimension of ICE as 1,000 by default. Here we show how the output dimension affects the results in Figure 3a. We can see that the average attack success rates on the six target models will rise when the output dimension is increased from 30 to 300, and will become stable when the output dimension > 300.

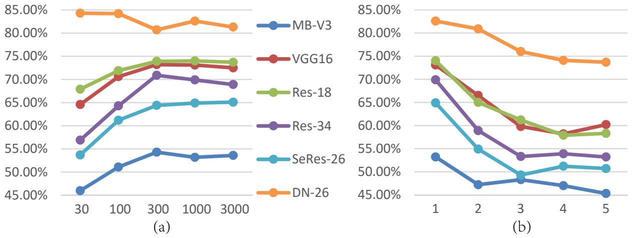

Figure 3: All experiments in this figure use Cifar-10 and ResNet-18 as the source dataset and the source model. (a): ICE's GTA results on six target models trained on Cifar-100 with different output dimensions. (b): ICE's GTA results on six target models trained on Cifar-100 with different numbers of T in the training phase.

4.5.4 Customized FGSM or Customized PGD?

Eq. 2 shows that we use Customized FGSM to perturb the input image in the inner-loop. We conduct an experiment to show why we use Customized FGSM instead of Customized PGD (Qin et al., 2021a). Customized PGD is a multi-step Customized FGSM. In this experiment, we increase the number of gradient ascent steps from 1 to 5. Figure 3b shows the experimental results. It is clear that the performances of ICE will be damaged by the increase of the number of gradient ascent steps. The possible reason for this phenomenon is that multi-step gradient ascent in the inner-loop makes the model $\mathcal{U}_\theta$ hard to be optimized in the outer-loop.

4.5.5 Using Entropy in outer-loop?

We use entropy to calculate the loss (Eq. 2) in the inner-loop of ICE but use cross-entropy (Eq. 3) in the outer-loop. Here we conduct another experiment to show the necessity of the cross-entropy in the outer-loop by replacing cross-entropy with entropy. We denote this version as ICE$_{en}$ and report its results in Table 4. The comparison between ICE and ICE$_{en}$ demonstrates that cross-entropy used in the outer-loop is necessary for ICE to achieve better performances.

5 Conclusion

In this paper, we propose the Generalized Transfer Attack (GTA) problem which is more challenging and more realistic than existing transfer attacks. To solve this novel problem, we modify some transferable adversarial attack methods and propose a novel Image Classification Eraser method. Experiments on several datasets demonstrate that existing transferable adversarial attack methods can be modified to tackle the GTA problem, and the proposed ICE performs the best on GTA.

6 Ethics Statement

Our work reveals the robustness issue of DNNs on the generalized transferable attack problem. The target models that use unseen label sets and unknown input shapes can be attacked by the proposed ICE method with high possibility. ICE is promising to evaluate the security of DNNs, and can be used to improve the robustness of DNNs, and has little potential negative societal impacts. Section A.8 and Figure 5 show that ICE tends to disturb images sampled from different categories with similar patterns, which may help researchers to better understand the robustness issue of DNN.

7 Reproducibility Statement

We provide our code in supplemental material and describe all the experimental settings in Section 4.2 and Appendix. The datasets we used are detailed in Sections 4 and A.2. The hyperparameter settings and the network structure are clear. The training of all source and target models are detailed in Section A.4, and the corresponding network architecture descriptions of these models can be found in Section A.4 and our code. The implementations of all baselines are described in Sections 4.1 and A.3. Overall, our work is easy to reproduce and follow.

References

Avishek Joey Bose, Gauthier Gidel, Hugo Berrard, Andre Cianflone, Pascal Vincent, Simon Lacoste-Julien, and William L Hamilton. Adversarial example games. *Advances in neural information processing systems*, 2020.

Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *international conference on learning representations*, 2018.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pp. 39–57. IEEE, 2017.

Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, pp. 15–26, 2017.

Minhao Cheng, Thong Le, Pin-Yu Chen, Jinfeng Yi, Huan Zhang, and Cho-Jui Hsieh. Query-efficient hard-label black-box attack: An optimization-based approach. *arXiv preprint arXiv:1807.04457*, 2018.

Minhao Cheng, Simranjit Singh, Patrick H. Chen, Pin-Yu Chen, Sijia Liu, and Cho-Jui Hsieh. Sign-opt: A query-efficient hard-label adversarial attack. In *international conference on learning representations*, 2020.

Shuyu Cheng, Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Improving black-box adversarial attacks with a transfer-based prior. pp. 10932–10942, 2019.

Francesco Croce and Matthias Hein. Minimally distorted adversarial examples with a fast adaptive boundary attack. In *International Conference on Machine Learning*, pp. 2196–2205. PMLR, 2020a.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International Conference on Machine Learning*, pp. 2206–2216. PMLR, 2020b.

Ambra Demontis, Marco Melis, Maura Pintor, Matthew Jagielski, Battista Biggio, Alina Oprea, Cristina Nita-Rotaru, and Fabio Roli. Why do adversarial attacks transfer? explaining transferability of evasion and poisoning attacks. *USENIX Security Symposium*, pp. 321–338, 2019.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 9185–9193, 2018.

Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4312–4321, 2019.

Jiawei Du, Hu Zhang, Tianyi Joey Zhou, Yi Yang, and Jiashi Feng. Query-efficient meta attack to deep neural networks. *International Conference on Learning Representations*, 2020.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pp. 1126–1135. PMLR, 2017.

Aditya Ganeshan, Vivek BS, and R Venkatesh Babu. Fda: Feature disruptive attack. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8069–8079, 2019.

Lianli Gao, Qilong Zhang, Jingkuan Song, Xianglong Liu, and Heng Tao Shen. Patch-wise attack for fooling deep neural network. In *European Conference on Computer Vision*, pp. 307–322. Springer, 2020.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *international conference on learning representations*, 2014.

Yiwen Guo, Qizhang Li, and Hao Chen. Backpropagating linearly improves transferability of adversarial examples. In *Advances in neural information processing systems 33 (NIPS 2020)*, 2020.

Zheng Haizhong, Zhang Ziqi, Gu Juncheng, Lee Honglak, and Prakash Atul. Efficient adversarial training with transferable adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1178–1187, 2020.

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.

Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1314–1324, 2019.

Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141, 2018.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

Qian Huang, Isay Katsman, Horace He, Zeqi Gu, Serge Belongie, and Ser-Nam Lim. Enhancing adversarial example transferability with an intermediate level attack. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4733–4742, 2019.

Zhichao Huang and Tong Zhang. Black-box adversarial attack with transferable model-based embedding. *International Conference on Learning Representations*, 2020.

Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In *International Conference on Machine Learning*, pp. 2137–2146. PMLR, 2018.

Khurram Javed and Martha White. Meta-learning representations for continual learning. *NeurIPS*, pp. 1818–1828, 2019.

Xu Kaidi, Liu Sijia, Zhao Pu, Chen Pin-Yu, Zhang Huan, Fan Quanfu, Erdogmus Deniz, Wang Yanzhi, and Lin Xue. Structured adversarial attack: Towards general implementation and better interpretability. *International Conference on Learning Representations*, 2019.

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

Alexey Kurakin, Ian Goodfellow, Samy Bengio, et al. Adversarial examples in the physical world, 2016.

Huichen Li, Xiaojun Xu, Xiaolu Zhang, Shuang Yang, and Bo Li. Qeba: Query-efficient boundary-based blackbox attack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1221–1230, 2020a.

Qizhang Li, Yiwen Guo, and Hao Chen. Practical no-box adversarial attacks against dnns. *Advances In Neural Information Processing Systems 2020*, 2020b.

Yingwei Li, Song Bai, Yuyin Zhou, Cihang Xie, Zhishuai Zhang, and Alan Yuille. Learning transferable adversarial examples via ghost networks. *AAAI*, pp. 11458–11465, 2020c.

Jiadong Lin, Chuanbiao Song, Kun He, Liwei Wang, and John E Hopcroft. Nesterov accelerated gradient and scale invariance for adversarial attacks. *International Conference on Learning Representations*, 2020.

Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. *international conference on learning representations*, 2019a.

Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. *international conference on learning representations*, 2017.

Zechun Liu, Haoyuan Mu, Xiangyu Zhang, Zichao Guo, Xin Yang, Kwang-Ting Cheng, and Jian Sun. Metapruning: Meta learning for automatic neural network channel pruning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3296–3305, 2019b.

Yantao Lu, Yunhan Jia, Jianyu Wang, Bai Li, Weiheng Chai, Lawrence Carin, and Senem Velipasalar. Enhancing cross-task black-box transferability of adversarial examples with dispersion reduction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Andriushchenko Maksym, Croce Francesco, Flammarion Nicolas, and Hein Matthias. Square attack: a query-efficient black-box adversarial attack via random search. *european conference on computer vision*, pp. 484–501, 2020.

Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. *international conference on learning representations*, 2018.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2574–2582, 2016.

Muhammad Muzammal Naseer, Salman H Khan, Muhammad Haris Khan, Fahad Shahbaz Khan, and Fatih Porikli. Cross-domain transferability of adversarial perturbations. *Advances in Neural Information Processing Systems*, 32:12905–12915, 2019.

Inkawhich Nathan, Kevin Liang J, Wang Binghui, Inkawhich Matthew, Carin Lawrence, and Chen Yiran. Perturbing across the feature hierarchy to improve standard and strict blackbox attack transferability. In *NIPS 2020*, 2020.

Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*, 2016.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pp. 506–519, 2017.

Yunxiao Qin, Weiguo Zhang, Zezheng Wang, Chenxu Zhao, and Jingping Shi. Layer-wise adaptive updating for few-shot image classification. *IEEE Signal Processing Letters*, 27:2044–2048, 2020.

Yunxiao Qin, Yuanhao Xiong, Jinfeng Yi, and Cho-Jui Hsieh. Training meta-surrogate model for transferable adversarial attack. *arXiv preprint arXiv:2109.01983*, 2021a.

Yunxiao Qin, Zitong Yu, Longbin Yan, Zezheng Wang, Chenxu Zhao, and Zhen Lei. Meta-teacher for face anti-spoofing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021b. doi: 10.1109/TPAMI.2021.3091167.

Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. Meta-learning for semi-supervised few-shot classification. *International Conference on Learning Representations*, 2018a.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *International Conference on Machine Learning*, pp. 4334–4343. PMLR, 2018b.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations*, 2015.

Gaurang Sriramanan, Sravanti Addepalli, Arya Baburaj, and Venkatesh R. Babu. Guided adversarial attack for evaluating and enhancing adversarial defenses. *Advances In Neural Information Processing Systems*, 2020.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, J. Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *international conference on learning representations*, 2014.

Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017.

Jonathan Uesato, Brendan O'Donoghue, Pushmeet Kohli, and Aaron van den Oord. Adversarial risk and the dangers of evaluating against weak attacks. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 5025–5034, 2018.

Lu Wang, Huan Zhang, Jinfeng Yi, Cho-Jui Hsieh, and Yuan Jiang. Spanning attack: reinforce black-box attacks with unlabeled data. *Machine Learning*, 109(12):2349–2368, 2020.

Xiaosen Wang and Kun He. Enhancing the transferability of adversarial attacks through variance tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1924–1933, 2021.

Xin Wang, Jie Ren, Shuyun Lin, Xiangming Zhu, Yisen Wang, and Quanshi Zhang. A unified approach to interpreting and boosting adversarial transferability. *International Conference on Learning Representations*, 2021.

Dongxian Wu, Yisen Wang, Shu-Tao Xia, James Bailey, and Xingjun Ma. Skip connections matter: On the transferability of adversarial examples generated with resnets. *international conference on learning representations*, 2020a.

Kaiwen Wu, Allen Wang, and Yaoliang Yu. Stronger and faster wasserstein adversarial attacks. *International Conference on Machine Learning*, pp. 10377–10387, 2020b.

Lei Wu, Zhanxing Zhu, Cheng Tai, et al. Understanding and enhancing the transferability of adversarial examples. *arXiv preprint arXiv:1802.09707*, 2018.

Weibin Wu, Yuxin Su, Xixian Chen, Shenglin Zhao, Irwin King, R. Michael Lyu, and Yu-Wing Tai. Boosting the transferability of adversarial samples via attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1158–1167, 2020c.

Zang Xiao, Xie Yi, Chen Jie, and Yuan Bo. Graph universal adversarial attacks - a few bad actors ruin graph learning models. *International Joint Conference on Artificial Intelligence*, pp. 3328–3334, 2021.

Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L Yuille. Improving transferability of adversarial examples with input diversity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2730–2739, 2019.

Zheng Yuan, Jie Zhang, Yunpei Jia, Chuanqi Tan, Tao Xue, and Shiguang Shan. Meta gradient adversarial attack. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

Chaoning Zhang, Philipp Benz, Adil Karjauv, and In So Kweon. Data-free universal adversarial perturbation and black-box attack. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7868–7877, 2021.

Wen Zhou, Xin Hou, Yongjun Chen, Mengyun Tang, Xiangqi Huang, Xiang Gan, and Yong Yang. Transferable adversarial perturbations. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 452–467, 2018.

A Appendix

A.1 Network architecture

As mentioned in Section 3 of the main body, the proposed ICE should be able to handle images with different resolutions because we cannot know the image shape in advance. Therefore, we build a fully convolutional neural network shown in Figure 4 as the backbone of ICE. The parameters M_1, M_2, M_3, and M_4 are set to 32, 64, 128, and 256, respectively.

A.2 More details of Tiered$_{T84}$ and Tiered$_{V56}$

For either Tiered$_{T84}$ or Tiered$_{V56}$, we use the image ID to rank all images of each category and use the first 1200 images of each category to compose the training set and use the last 100 images to compose the testing set. For example, the IDs of the two images 'n01530575_5.JPEG' and 'n01530575_23.JPEG' from the 'n01530575' category are 5 and 23.

A.3 Additional implementation details of baselines

Some implementation details of baselines have been introduced in the main body. Here we introduce the additional implementation details of baselines.

MI: Parameter μ of MI is set to 1.

DI:. We use the code[1] to implement DI in all our experiments. We set 'FLAGS.image_resize' to 36, 64, or 96, when the resolution of the input image is 32, 56, or 84, respectively. The input diverse possibility p is set to 1.0.

TI-DIM:. We use the code[2] to implement TI-DIM in all our experiments.

IR:. We use the code[3] to implement IR in all our experiments. The hyper-parameter 'args.grid_scale' and 'args.sample_grid_num' are set to 1 and 16, respectively, for all experiments.

AEG. We implement AEG in our experiment by referring to the code[4]. Given each source dataset D_k and the corresponding source models trained on it, we adversarially train a perturbation generator together with a critic. The generator can be denoted as G_k. For example, for the experiment '-Cifar-10', we train three generators on the three datasets Cifar-100, Tiered$_{T84}$ and Tiered$_{V56}$, and denote them as G_1, G_2, G_3, respectively. The architecture of all generators is the encoder-decoder defined in Tab.7 of AEG's paper. Note that considering ground-truth label is unavailable in inference, we do not use the label as the additional input signal for the decoder when training the generators. Each generator's input-size is the same with the image-shape of the training dataset. On either Cifar-10 or Cifar-100, we train the generator and the critic for 500 epochs with the learning rate of 0.001. On either Tiered$_{T84}$ or Tiered$_{V56}$, we train the generator and the critic for 120 epochs with the learning rate of 0.001.

In the inference phase, we use the following steps to evaluate AEG on generalized transferable attack. 1) Resize the testing clean image x to the input shapes of all generators and then feed the resized images to all generators, respectively. 2) Obtain the perturbations generated by the generators, which can be formulated as $\delta_k = G_k\big(\text{resize}(x, \text{resolution}(G_k))\big)$. 3) Average fuse all the generated perturbations with the formulation $\delta = \frac{1}{m}\sum_{s=1}^{m}\delta_k$. 4) Obtain the adversarial example $\hat{x} = x + \epsilon \cdot \text{sign}(\delta)$, where $\epsilon = 15$. 5) Feed x and $\hat{x}$ into the unknown target model $\mathbb{M}$, and get the predictions. 6) The generalized transferable attack is successful if $\mathbb{M}(x) \neq \mathbb{M}(\hat{x})$.

MTA. We refer to the MTA paper to implement it on the generalized transferable attack problem. Given each source dataset D_k and the corresponding source models trained on it, we train a meta-surrogate model, which can be demoted as $\mathcal{S}_k$. Each meta-surrogate model's input-size is the same with the image-shape of the training dataset. We train the meta-surrogate models on all the source

[1] https://github.com/cihangxie/DI-2-FGSM

[2] https://github.com/dongyp13/Translation-Invariant-Attacks

[3] https://github.com/xherdan76/A-Unified-Approach-to-Interpreting-and-Boosting-Adversarial-Transferability

[4] https://github.com/joeybose/Adversarial-Example-Games

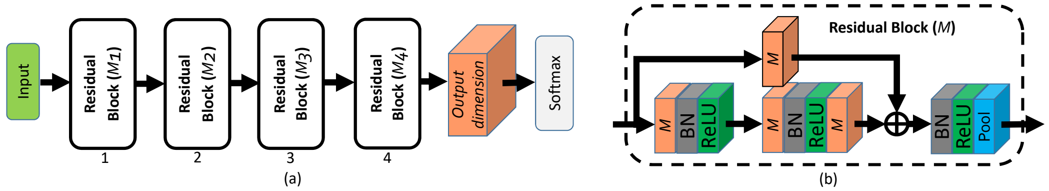

Figure 4: (a) The network architecture of the proposed ICE. It is composed of four cascaded residual blocks, one convolutional layer, and one softmax layer. (b) The inner structure of the residual block. Orange cube denotes convolutional layer and the number on it denotes the number of filters of the convolutional layer. 'Pool' in the last residual block is global average pooling and 'Pool' in all the other residual blocks are max-pooling with both stride and pooling size set to 2.

datasets with the following settings. On either Cifar-10 or Cifar-100, we train the meta-surrogate model for 50,000 iterations with the parameter ϵ_c and number of attack steps T_t in Customized PGD set to 1600 and 7, respectively. On either Tiered$_{T84}$ or Tiered$_{V56}$, we train the meta-surrogate model for 70,000 iterations with the parameter ϵ_c and number of attack steps T_t set to 2100 and 4, respectively. On each training dataset, ϵ_c is exponentially decayed by 0.9× for every 4000 iterations. The learning rate and the batch size are set to 0.001 and 64, respectively.

We use the following steps to evaluate MTA on generalized transferable attack. 1) Resize the testing clean image x to the input shapes of all meta-surrogate models and feed the resized images to all meta-surrogate models, respectively. Then the inference of x on the meta-surrogate model $\mathcal{S}_{D_k}$ can be formulated as $y_{D_k}^{(0)} = \mathcal{S}_{D_k}(x_{D_k}^{(0)})$, where $x_{D_k}^{(0)} = \text{resize}(x, \text{resolution}(\mathcal{S}_{D_k}))$. 2) Because we cannot access the category of the image x in advance, no ground-truth label can be leveraged to perturb the resized image. Therefore, for each meta-surrogate model, we generate adversarial perturbation by maximizing the entropy (as used in our ICE) for T gradient ascent steps. The i-th step can be formulated as

$$\begin{cases} y_{D_k}^{(i-1)} = \mathcal{S}_{D_k}(x_{D_k}^{(i-1)}), \\ \delta_{D_k}^{(i-1)} = \text{sign}\big(\nabla_{x_{D_k}^{(i-1)}} \mathcal{L}(y_{D_k}^{(i-1)})\big), \\ x_{D_k}^{(i)} = \text{clip}(x_{D_k}^{(i-1)} + \frac{\epsilon}{T} \cdot \delta_{D_k}^{(i-1)}), \end{cases} \tag{7}$$

where $y_{D_k}^{(i-1)}$ is the meta-surrogate model's output and $x_{D_k}^{(i)}$ and $\delta_{D_k}^{(i-1)}$ are the adversarial example and the perturbation generated in the i-th step, respectively. 3) Resize all the adversarial examples to the original shape of the image x and average fuse all adversarial examples to one image x_{adv} following the formulation $x_{adv} = \frac{1}{m} \cdot \sum_{k=1}^{m} \cdot \big(\frac{1}{N_k} \cdot \sum_{j=1}^{N_k} \text{resize}(x_{D_k}^{(T)}, \text{resolution}(x))\big)$. 4) Generate adversarial example $\hat{x}$ by the formulation $\hat{x} = \text{clip}(x + \epsilon \cdot \text{sign}(x_{adv} - x))$, which is the SP step defined in Section 3.2 of the main-body. 5) Feed the adversarial example $\hat{x}$ and the clean image x into the unknown target model $\mathbb{M}$ and get its predictions. 6) The GTA process is successful if $\mathbb{M}(\hat{x}) \neq \mathbb{M}(x)$.

A.4 Training details of source and target models

On each of the dataset Cifar-10, Cifar-100, Tiered$_{T84}$, and Tiered$_{V56}$, we train the seven models ResNet-18, -34, SeResNet-26, VGG-16, MobileNet-V1, MobileNet-V3, and DenseNet-26. The network architectures of all the seven models are defined in the public GitHub repository[5]. We use consistent hyper-parameters to train all the models for 80,000 iterations without data augmentation. The learning rate, L2 weight decay, and batch size are set to 0.01, 1e-5, and 128, respectively. Table 5 shows the seven models' accuracies on the four datasets.

[5] https://github.com/yxlijun/cifar-tensorflow

Table 5: Accuracies of source and target model on the four datasets.

Dataset	Mobile-Net-V1	MobileNet-V3	VGG-16	ResNet-18	ResNet-34	SeResNet-26	DenseNet-26
Cifar-10	82.0%	80.0%	92.9%	91.8%	92.6%	88.3%	91.2%
Cifar-100	48.0%	43.9%	68.5%	68.0%	69.3%	61.7%	64.6%
Tiered$_{T84}$	38.1%	35.3%	46.9%	47.0%	49.9%	48.9%	44.6%
Tiered$_{V56}$	34.9%	32.1%	46.3%	45.9%	48.1%	43.2%	46.0%

A.5 Additional experiments

A.5.1 Smaller perturbation

Here we report another generalized transferable attack experiment on Cifar-100. In this experiment, we test whether the proposed ICE is sensitive to the perturbation scale by changing ϵ from the default 15 to 8. The source datasets are Cifar-10, Tiered$_{T84}$, and Tiered$_{T56}$. The source models are three ResNet-18 models respectively trained on the three source datasets. Table 6 shows the experimental results. It is observed that the ICE outperforms baselines with a significant margin. In lots of cases, the attack success rate of ICE is more than twice as much as that of baselines, which further indicates the effectiveness of the proposed ICE.

Table 6: GTA success rates on Cifar-100 with ϵ=8.

Resource	Method	MobileNet-V3	VGG-16	ResNet-18	ResNet-34	SeResNet-26	DenseNet-26
Cifar-10 + Tiered$_{T84}$ + Tiered$_{V56}$ (ResNet-18)	FGSM	18.3%	29.9%	22.3%	22.1%	24.2%	39.7%
	PGD	16.4%	25.9%	19.3%	18.8%	21.3%	36.0%
	DI	16.5%	25.8%	19.3%	18.9%	21.5%	36.1%
	MI	17.5%	28.5%	21.6%	21.2%	23.7%	39.5%
	TI-DIM	25.6%	20.1%	21.3%	21.8%	21.3%	27.6%
	IR	21.0%	30.6%	25.5%	24.9%	25.0%	42.2%
	AEG	21.1%	29.7%	23.5%	22.8%	27.6%	38.0%
	MTA	17.2%	26.5%	21.0%	21.3%	23.8%	35.3%
	ICE	**26.9%**	**49.9%**	**62.4%**	**58.2%**	**46.1%**	**72.2%**

A.5.2 Attacking robust models

We performed a new experiment to conduct generalized transfer attack on robust models. In this experiment, we use ResNet-18 and Cifar-10 as the source model and the source dataset, and disturb the images from Cifar-100. The target models are adversarially trained ResNet-18 and ResNet-34 on Cifar-100, which can be denoted as ResNet-18$_{adv}$ and ResNet-34$_{adv}$, respectively. To obtain ResNet-18$_{adv}$, we firstly use the normally trained ResNet-18 to generate adversarial examples for all training examples with FGSM (ϵ=15), and then retrain ResNet-18 on all the clean training images and the adversarial images. We obtain ResNet-34$_{adv}$ in a similar way. The two models finally achieve approximately 46.9% and 47.3% testing accuracies on Cifar-100. The generalized transfer attack results on ResNet-18$_{adv}$ and ResNet-34$_{adv}$ are reported in Table 7. Obviously, though all the methods perform not well to attack adversarially trained models, ICE can still show its advantage in this experiment.

Table 7: GTA success rates on robust Cifar-100 models.

Resource	Method	ResNet-18$_{adv}$	ResNet-34$_{adv}$
Cifar-10 (ResNet-18)	FGSM	11.1%	12.9%
	PGD	9.7%	10.1%
	DI	11.0%	11.6%
	MI	10.8%	11.8%
	TI-DIM	16.7%	15.9%
	IR	13.8%	13.1%
	ICE	**16.9%**	**17.5%**

A.5.3 Re-implement PGD-based baselines with KL divergence

In Section 4.3, the PGD-based baselines disturb input images by maximizing the entropy loss of source models. In Section 4.5.2, the PGD-based baselines disturb input images by maximizing the cross-entropy loss of source models, where the perturbation noise is formulated as $\delta = \text{sign}\big(\nabla_{x^{(i-1)}}\text{CE}(y, \text{argmax}(y))\big)$. argmax$(y)$ is regarded as the pseudo label for the input. Here we re-implement PGD-based baselines with KL divergence, which means the baselines disturb the input images by maximizing the KL divergence between the predict distribution y and y_0, where y_0 is the distribution predict for the original clean image x. The perturbation generated in each gradient ascent step can be reformulated as $\delta = \text{sign}\big(\nabla_{x^{(i-1)}}\text{KL}(y, y_0)\big)$. In this experiment, we use ResNet-18 and Cifar-10 as the source model and source dataset, and use Cifar-100 as the target dataset. The generalized transfer attack results on MobileNet-V3, VGG-16, ResNet-18, ResNet-34, SeResNet-26, and DenseNet-16 are reported in Table 8. The comparison between the results here and those in Table 2 demonstrate that the KL divergence cannot improve baselines.

Table 8: GTA success rates on Cifar-100 with KL divergence.

Resource	Method	MobileNet-V3	VGG-16	ResNet-18	ResNet-34	SeResNet-26	DenseNet-26
Cifar-10 (ResNet-18)	FGSM	49.3%	63.3%	57.9%	56.5%	58.8%	72.5%
	PGD	39.6%	52.5%	45.0%	44.5%	48.0%	62.7%
	DI	41.2%	54.6%	46.5%	45.9%	50.0%	64.2%
	MI	47.3%	62.6%	56.9%	56.1%	58.2%	72.4%
	TI-DIM	50.6%	43.1%	45.6%	45.9%	46.9%	52.4%

A.5.4 Optimize a single adversarial image

In our previous experiments, we implement PGD-based baselines with the pipeline introduced in Section 4.1. Here we re-implement PGD-based baselines with a new pipeline. The new pipeline uses T gradient ascent iterations to disturb the input image, and the i-th iteration contains the following three steps.

1) We resize the image $x^{(i-1)}$ to the input shapes of all source models and then feed the resized images to the source models, respectively. $x^{(i-1)}$ is the perturbed image generated in the $i-1$ iteration and $x^{(0)} = x$. Then the inference of $x^{(i-1)}$ on one source model $\mathbf{M}^j_{D_k}$ can be formulated as $y^j_{D_k} = \mathbf{M}^j_{D_k}(x^j_{D_k})$, where $x^j_{D_k} = \text{resize}(x^{(i-1)}, \text{resolution}(\mathbf{M}^j_{D_k}))$ is the resized image for the source model $\mathbf{M}^j_{D_k}$.
2) We calculate the prediction losses of all the source models with the formulation $L = \frac{1}{m} \cdot \sum_{k=1}^{m} \cdot \big(\frac{1}{N_k} \cdot \sum_{j=1}^{N_k} \mathcal{L}(y^j_{D_k})\big)$, where $\mathcal{L}$ is entropy.
3) We disturb the input image $x^{(i-1)}$ with the formulation $x^{(i)} = \text{clip}(x^{(i-1)} + \frac{\epsilon}{T} \cdot \delta)$, where $\delta = \text{sign}\big(\nabla_{x^{(i-1)}} L\big)$.

After T iterations, we obtain the disturbed image $x^{(T)}$, and feed it and the clean image x into the unknown target model $\mathbb{M}$ and get its predictions. The GTA process is successful if $\mathbb{M}(x^{(T)}) \neq \mathbb{M}(x)$.

We perform another experiment to evaluate how the re-implemented PGD-based baselines perform on GTA. In this experiment, Cifar-10, Tiered$_{T84}$, and Tiered$_{V56}$ are used as source datasets, and three ResNet-18 respectively trained on the three datasets are used as the source models. The experimental results on the target models MobileNet-V3, VGG-16, ResNet-18, ResNet-34, SeResNet-26,and DenseNet-26 are reported in Table 9. Compared with the results in the last row of Table 2, optimizing a single image improves FGSM and MI, but damages DI and TI-DIM.

Table 9: GTA success rates of the PGD-based baselines re-implemented with new pipeline. Target dataset is Cifar-100.

Resource	Method	MobileNet-V3	VGG-16	ResNet-18	ResNet-34	SeResNet-26	DenseNet-26
Cifar-10 + Tiered$_{T84}$ + Tiered$_{V56}$ (ResNet-18)	FGSM	47.5%	63.2%	56.3%	55.5%	58.2%	72.6%
	PGD	38.8%	53.6%	43.1%	43.1%	47.5%	64.6%
	DI	38.6%	53.9%	43.3%	42.8%	47.7%	64.7%
	MI	46.2%	62.8%	53.8%	52.7%	56.6%	71.9%
	TI-DIM	47.9%	43.8%	44.9%	45.1%	44.2%	54.0%

A.5.5 Attacking Fine-Grained classification models

We performed a novel experiment to evaluate whether the proposed method can be used to disturb the images from fine-grained classification dataset. In this experiment, the source dataset and source model are Cifar-10 and ResNet-18 respectively, and the target dataset is CUB, which is a fine-grained image classification dataset and differs greatly from Cifar-10. The target models are ResNet-18 and DenseNet-26 trained on CUB. The two target models use a consistent input resolution of 112x112 and achieve approximately 54.9% and 48.3% accuracies. The attack success rates on the two target models are reported in Table 10.

Table 10: GTA success rates on CUB models.

Resource	Method	ResNet-18	DenseNet-26
	FGSM	22.7%	32.6%
	PGD	17.2%	10.1%
Cifar-10	DI	49.2%	72.5%
(ResNet-18)	MI	22.0%	30.7%
	TI-DIM	62.8%	73.4%
	ICE	**65.1%**	**87.0%**

A.5.6 Using Knowledge distillation to train a single model

In this experiment, we use Cifar-10, Tiered$_{T84}$, and Tiered$_{V56}$ as source datasets, and use three ResNet-18 respectively trained on the source datasets as source models. First, we build another model G that has three heads corresponds to the three source datasets. The backbone of G is the same with that of the proposed ICE. The first head that has 10 output nodes is used to classify Cifar-10 images. The second head which has 351 output nodes is used to classify Tiered$_{T84}$ images. The third head which has 257 output nodes is used to classify Tiered$_{V56}$ images. Second, we use the three source models and three datasets to train G with offline knowledge distillation. The model G obtain test accuracies of 97.5%, 61.7%, and 70.3% on Cifar-10, Tiered$_{T84}$, and Tiered$_{V56}$, respectively. Finally, we use the model G to disturb the images from Cifar-100. The GTA success rates on the target models MobileNet-V3, VGG-16, ResNet-18, ResNet-34, SeResNet-26, and DenseNet-26 are reported in Table 11.

Table 11: GTA success rates when using a single trained model. Target dataset is Cifar-100.

Resource	Method	MobileNet-V3	VGG-16	ResNet-18	ResNet-34	SeResNet-26	DenseNet-26
Cifar-10	FGSM	46.4%	59.6%	51.0%	51.4%	55.1%	70.6%
+ Tiered$_{T84}$	PGD	36.9%	49.6%	40.3%	40.1%	43.4%	61.3%
+ Tiered$_{V56}$	MI	43.9%	56.6%	47.6%	47.3%	52.4%	68.4%
(ResNet-18)	TI	49.2%	43.7%	45.9%	46.5%	44.6%	54.4%

A.5.7 Comparison between ICE and UAP

Here we implement UAP (Zhang et al., 2021) and compare it with the proposed ICE. In this experiment, we use Cifar-10 and ResNet-18 as the source dataset and the source model, and optimize the universal adversarial perturbation by minimizing the cosine similarity between $f(x)$ and $f(x+\delta)$, where f is the ResNet-18 model. x and δ are the clean image and the perturbation. Then we use the trained δ to disturb the images from Cifar-100 and attack Cifar-100 models. L_∞ perturbation scale of δ is set to 15/255, which is consistant to the default setting in our work. The attack success rates on the target models MobileNet-V3, VGG-16, ResNet-18, ResNet-34, SeResNet-26, and DenseNet-26 are reported in Table 12. It is obvious that ICE outperforms UAP in most testing scenes.

Table 12: Comparison between UAP and ICE. Target dataset is Cifar-100.

Resource	Method	MobileNet-V3	VGG-16	ResNet-18	ResNet-34	SeResNet-26	DenseNet-26
Cifar-10	UAP	53.6%	66.8%	68.2%	**66.5%**	59.3%	77.0%
(ResNet-18)	**ICE**	**54.9%**	**67.3%**	**71.5%**	64.0%	**61.2%**	**83.5%**

A.6 COMPUTATIONAL COST

We conduct all experiments on Tesla P40 GPU. The training cost of the proposed ICE is determined by the batch size, the backbone, the used source datasets, the source models, and *etc.*. The number of parameters of ICE is determined by the backbone. The parameter settings have been introduced in Section 4.2 in the main-body. The backbone which contains about 7.7M parameters is shown in Figure 4. With the source datasets Cifar-10, Tiered$_{T84}$, and Tiered$_{56}$, and with the three corresponding ResNet-18 source models, training the model $\mathcal{U}_\theta$ costs about 4.9T FLOPs per iteration. With the source datasets Cifar-10 and Tiered$_{T84}$, and with the two corresponding ResNet-18 source models, training the model $\mathcal{U}_\theta$ costs about 3.5T FLOPs per iteration.

In inference, the cost of ICE is determined by the backbone, the size of the testing image, and the number of gradient ascent steps T, which is set to 10 in our work. The inference cost of PGD-based baselines depends on the size of the testing image, the source models, and the number of gradient ascent steps T. When the testing image comes from Cifar-100, ICE costs approximately 1.2G FLOPs per gradient ascent step per image, while the PGD-based baseline MI, DI, or TI-DIM cost approximately 10.2G FLOPs, which indicates that ICE is much more efficient than PGD-based baselines in inference.

A.7 MORE ANALYSE ABOUT SP

The ablation study shown in Section 4.5.1 demonstrates that the trick SP in this paper is important for all methods to achieve better generalized transferable attack success rates. It is also observed that without SP, MI performs the best among all the methods. This is because MI utilizes gradient momentum to improve the attack success rate, and the momentum will enlarge the average perturbation scale of each pixel while keeping the L_∞ of the perturbation map unchanged, which plays a similar role to SP. For instance, without SP, the average perturbation scale of each pixel of the adversarial examples generated via MI is approximately 11. As a comparison, for the other PGD-based methods and ICE, the average perturbation scale of each pixel of the generated adversarial examples is no more than 9.

A.8 VISUALIZATION

We visualize some generated adversarial examples and the corresponding noise maps in Figure 5. The corresponding clean images are sampled from Tiered$_{T84}$, and the source datasets are Cifar-10, Cifar-100, and Tiered$_{V56}$, and the source models are three ResNet-18 respectively trained on the source datasets. It is very interesting to see that the perturbation noise generated by ICE differs greatly from those generated by the other methods. The possible reason for this phenomenon is that ICE is trained to maximize the prediction entropy without using labels. In other words, ICE needs to disturb the input image without knowing the ground-truth classification information, which forces ICE to learn some perturbation pattern that correlates little with ground-truth information.

In Figure 5, we can see that ICE generates similar perturbation noises for different input images, which means ICE may learn a perturbation pattern that is generalizable across different image categories. In our opinion, learning a generalizable perturbation pattern may is a straightforward way for ICE to solve generalized transfer attack. The visualization may remind us that it is possible to find a universal perturbation pattern that can solve generalized transfer attack, which can be regarded as another contribution of ICE. The visualization may also help us to understand and improve the robustness of DNNs.

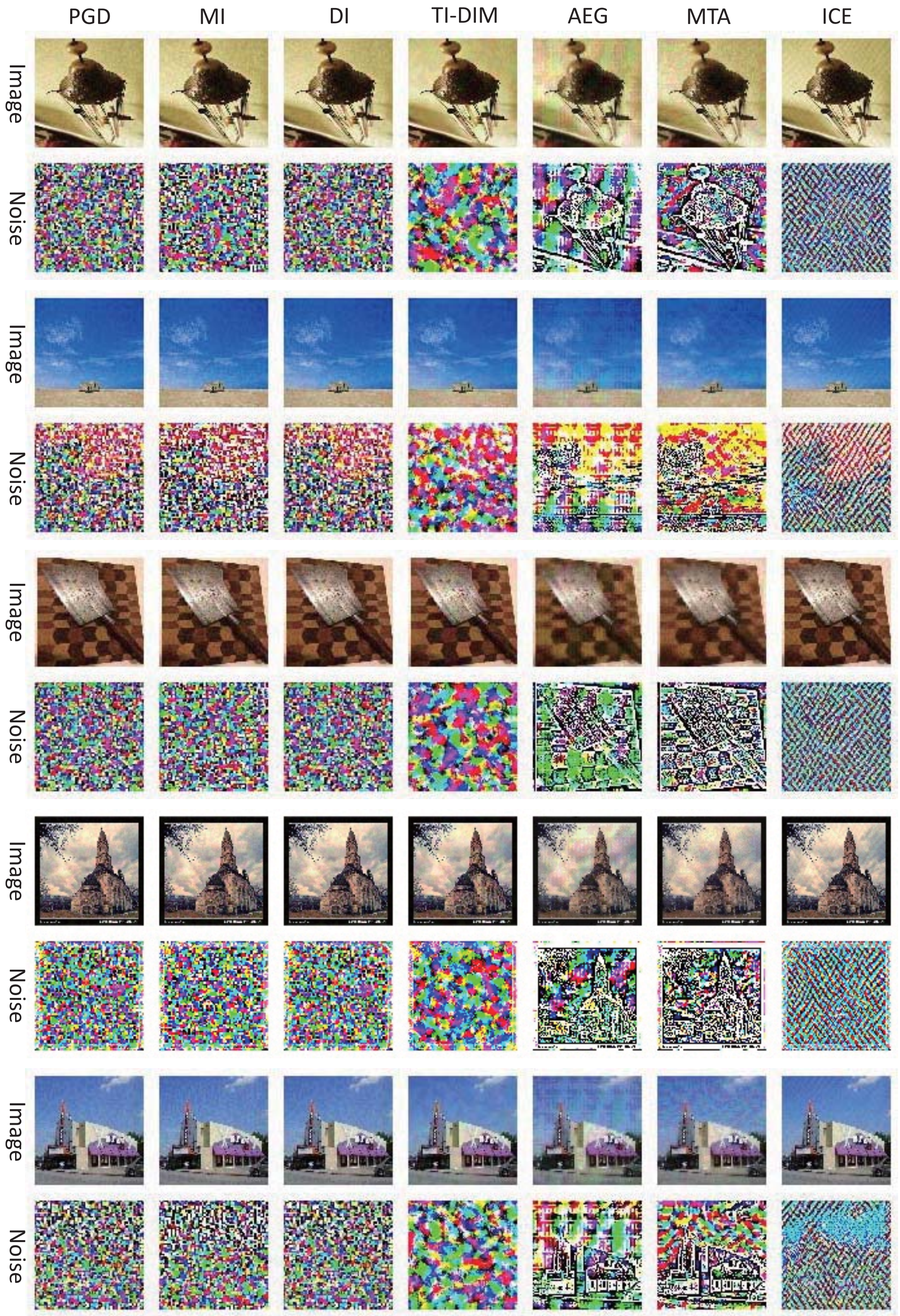

Figure 5: Some adversarial examples for clean images from Tiered$_{T84}$. The adversarial examples are generated via PGD, MI, DI, TI-DIM, AEG, MTA, and the proposed ICE with $\epsilon = 15$.