Towards Transferable Targeted Adversarial Examples

Zhibo Wang[†,§,*], Hongshan Yang[†], Yunhe Feng[‡], Peng Sun[♯], Hengchang Guo[ß], Zhifei Zhang[≀], Kui Ren[†]
[†]School of Cyber Science and Technology, Zhejiang University, P. R. China
[§]ZJU-Hangzhou Global Scientific and Technological Innovation Center
[‡]Department of Computer Science and Engineering, University of North Texas, USA
[♯]College of Computer Science and Electronic Engineering, Hunan University, P. R. China
[ß]School of Cyber Science and Engineering, Wuhan University, P. R. China [≀]Adobe Research
{zhibowang, yanghs, kuiren}@zju.edu.cn, yunhe.feng@unt.edu
psun@hnu.edu.cn, hc_guo@whu.edu.cn, zzhang@adobe.com

Abstract

Transferability of adversarial examples is critical for black-box deep learning model attacks. While most existing studies focus on enhancing the transferability of untargeted adversarial attacks, few of them studied how to generate transferable targeted adversarial examples that can mislead models into predicting a specific class. Moreover, existing transferable targeted adversarial attacks usually fail to sufficiently characterize the target class distribution, thus suffering from limited transferability. In this paper, we propose the Transferable Targeted Adversarial Attack (TTAA), which can capture the distribution information of the target class from both label-wise and feature-wise perspectives, to generate highly transferable targeted adversarial examples. To this end, we design a generative adversarial training framework consisting of a generator to produce targeted adversarial examples, and feature-label dual discriminators to distinguish the generated adversarial examples from the target class images. Specifically, we design the label discriminator to guide the adversarial examples to learn label-related distribution information about the target class. Meanwhile, we design a feature discriminator, which extracts the feature-wise information with strong cross-model consistency, to enable the adversarial examples to learn the transferable distribution information. Furthermore, we introduce the random perturbation dropping to further enhance the transferability by augmenting the diversity of adversarial examples used in the training process. Experiments demonstrate that our method achieves excellent performance on the transferability of targeted adversarial examples. The targeted fooling rate reaches 95.13% when transferred from VGG-19 to DenseNet-121, which significantly outperforms the state-of-the-art methods.

*Zhibo Wang is the corresponding author.

1. Introduction

As an important branch of artificial intelligence (AI), deep neural networks (DNNs) contribute to many real-life applications, *e.g.*, image classification [1, 2], speech recognition [3, 4], face detection [5, 6], automatic driving technology [7], *etc*. Such broad impacts have motivated a wide range of investigations into the adversarial attacks on DNNs, exploring the vulnerability and uncertainty of DNNs. For instance, Szegedy *et al*. showed that DNNs could be fooled by adversarial examples crafted by adding human-indistinguishable perturbations to original inputs [8]. As successful adversarial examples must be imperceptible to humans but cause DNNs to make a false prediction, how to design adversarial attacks to generate high-quality adversarial examples in a general manner remains challenging.

Adversarial attack methods can be divided into untargeted attacks and targeted attacks. Untargeted attacks try to misguide the model to predict arbitrary incorrect labels, while targeted adversarial attacks expect the generated adversarial examples can trigger the misprediction for a specific label. The transferability of adversarial examples is crucial for both untargeted and targeted attacks, especially in black-box attack scenarios where the target model is inaccessible. However, most existing studies focus on enhancing the transferability of untargeted adversarial attacks, through data augmentation [9, 10], model aggregation [11,12], feature information utilization [13–15], or generative methods [16–18]. Although some of them could be extended to the targeted adversarial attacks by simply modifying the loss function, they could not extract sufficient transferable information about the target class due to the overfitting of the source model and the lack of distribution information of the target class, thus demonstrating limited transferability.

Some recent works [15, 19–21] studied how to boost the transferability of targeted adversarial attacks by learning the target class information from either the label or the feature perspective, leveraging the label probability distribution and feature maps respectively. The label-wise information, often output by the last layer of the classification model, can effectively reflect the direct correlation between image distribution and class labels. However, learning with just the label-wise information is proven to retain high-level semantic information of the original class [22], thus leading to low cross-model transferability. The feature-wise information, which can be obtained from the intermediate layer of the classification model, has been proven [23] to have transferability due to the mid-level layer of different DNNs following similar activation patterns. However, the feature-wise information is not sensitive to the target label and fails to trigger the targeted misclassification.

In summary, only the information extracted from the label or the feature cannot generate high-transferability targeted adversarial examples. Meanwhile, most existing targeted attacks assume that the training dataset of the target model (*i.e.*, target domain) is accessible and could be leveraged by the attackers to train a shadow model in the target domain as the simulation of the target model, which however is not practical in realistic scenarios.

In this paper, we aim to generate highly transferable targeted adversarial examples in a more realistic but challenging scenario where the attacker cannot access the data of the target domain. To this end, we are facing two main challenges. The first challenge is ***how to generate targeted adversarial examples with both cross-model and cross-domain transferability?*** Cross-domain images usually vary significantly in characteristic distribution (e.g., have different attributes even labels), making it difficult for models to transfer the knowledge to unknown domains. Besides, when involved models adopt different architectures, domain knowledge learned by the source model becomes less transferable, resulting in more difficulties in cross-model and cross-domain transferable adversarial attacks. The second challenge is ***how to improve the transferability of targeted adversarial attacks?*** As the targeted adversarial attack needs to distort the ground-truth label and trigger the model to predict the target label simultaneously, causing the transferability of the targeted attack hard to achieve due to the dual objective of obfuscating original class information and recognizing target class information.

To solve such challenges, we propose Transferable Targeted Adversarial Attack (TTAA) to generate highly transferable targeted adversarial examples. The main idea of our method is to capture the distribution information of the target class from both label-wise and feature-wise perspectives. We design a generative adversarial network, which generates targeted adversarial examples by a generator and captures the distribution information of the target class by label-feature dual discrimination, consisting of the label discriminator and the feature discriminator. More specifically, the label discriminator learns the label-related distribution information from the label probability distribution and the feature discriminator extracts transferable distribution information of the target class via feature maps output by the intermediate layer of the label discriminator. Meanwhile, we propose random perturbation dropping to enhance our training samples, which applies random transformations to augment the diversity of the adversarial examples used during the training process to improve the robustness of the distribution information of the target class on the adversarial examples. In generative adversarial training, such distribution information extracted by discriminators guides the generator to acquire the label-related and transferable distribution information of the target class. Therefore the targeted adversarial examples achieve both high cross-model and cross-domain transferability.

Our main contributions are summarized as follows.

- We propose a general framework for targeted adversarial attack that works in both non-cross-domain and cross-domain scenarios. This framework could be easily integrated with other targeted adversarial attacks to improve their cross-model and cross-domain targeted transferability.
- Our proposed Transferable Targeted Adversarial Attack could extract the target class distribution from feature-wise and label-wise levels to promote perturbations to acquire label-related and transferable distribution information of the target class, thus generating highly transferable targeted adversarial examples.
- Extensive experiments on diverse classification models demonstrate the superior targeted transferability of adversarial examples generated by the proposed TTAA as compared to state-of-the-art transferable attacking methods, no matter whether the attack scenario is cross-domain or non-cross-domain.

2. Related Work

Since Szegedy *et al*. [8] demonstrated the existence of adversarial examples, many adversarial attack methods [11–13, 24–28] have been proposed to improve the transferability of adversarial examples. We divide these transferable adversarial attack methods into two categories via attack objective, namely transferable untargeted adversarial attacks and transferable targeted adversarial attacks.

Transferable untargeted adversarial attacks aim to deceive the target model to output incorrect results no matter what the misclassifications are. Dong *et al*. [10] integrated the momentum into the iterative process for getting stable update directions and avoiding poor local optimum, thus resulting in more transferable adversarial examples. Xie *et al*. [9] created diverse input patterns by adding random transformations into the input images at each iteration to enhance the transferability. Poursaeed *et al*. [16] proposed generative models for creating universal perturbation and image-independent perturbation to improve transferability. Naseer *et al*. [17] crafted adversarial examples with relative cross-entropy loss function, which enables domain-invariant perturbations and launches transferable adversarial attacks. Wang *et al*. [14] utilized aggregated gradients to disrupt important object-aware features that dominate the model decision to enhance transferability. Zhang *et al*. [18] leveraged a generative model to disrupt low-level features of input image extracted by a pre-trained model based on the ImageNet dataset, enhancing the transferability of the adversarial examples. Zhu *et al*. [29] optimized both the model gradient and data distribution gradient to directly push the images away from their original distribution, thereby boosting the transferability. Those untargeted adversarial attack methods would suffer from low transferability when performing targeted adversarial attacks.

Transferable targeted adversarial attacks aim to misguide the target model to predict specific results, *i.e*., the target label. Inkawhich *et al*. [19, 20] applied one or more auxiliary neural networks to learn the layer-wise deep feature distribution of the target class to improve the targeted transferability. Gao *et al*. [15] measured similarities between feature maps by high-order statistics with translation invariance to obtain transferable targeted adversarial attacks. Naseer *et al*. [21] aligned the global distribution and local neighborhood structure between the source domain and target, enhancing the targeted transferability. Byun *et al*. [30] diversified the input through differentiable rendering 3D objects to improve the transferability of the targeted adversarial examples. Those transferable targeted adversarial attacks have limited transferability because they only utilize feature or label information and cannot adequately characterize the target class information. Different from existing works, our method leverages label-feature dual distribution information of the target class to craft adversarial examples with high cross-model transferability and cross-domain transferability.

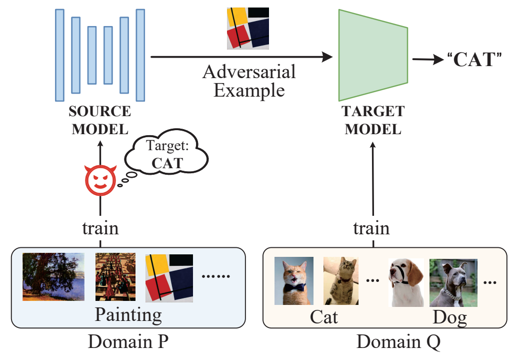
3. Targeted Adversarial Attack

Given a deep learning classification model $f(x) : \mathbb{X} \rightarrow \mathbb{Y}$, where $\mathbb{X}$ and $\mathbb{Y}$ denote the images and labels, respectively. Let $x \in \mathbb{X}$ denote an image from the image domain $\mathbb{X}$, which is of the size of $\mathbb{R}^{H\times W\times D}$ with H, W, D denoting the height, width, and depth of the image, respectively, and $y \in \mathbb{Y}$ denote the ground truth label of the image x.

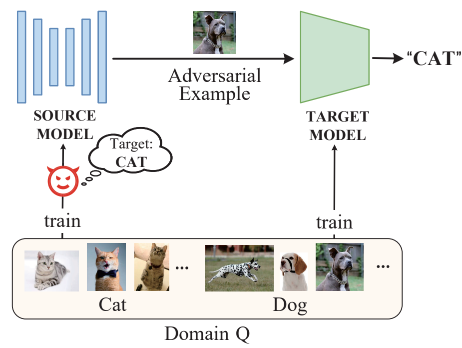
Targeted adversarial attack aims to craft the perturbation $\delta \in \mathbb{R}^{H\times N\times W}$ that misguides the prediction result of the classifier $f(\cdot)$ from the ground truth label y to the targeted label y_t. To ensure the indistinguishability of the perturbation, ℓ_p-norm is used to regularize the perturbation δ to the range ϵ. The optimization goal of the targeted adversarial attack is as follows.

$$\min_{\delta} \ \ell(f(x+\delta), y_t), \ \ s.t. ||\delta||_p \leq \epsilon, \tag{1}$$

where the loss function $\ell(\cdot)$ is adopted to measure the distance between the predicted result of the adversarial example $f(x+\delta)$ and the target label y_t. The final targeted adversarial example is generally the summation of the original image x and the perturbations δ, *i.e*., $x^{adv} = x + \delta$.



(a) Cross-Domain Targeted Attack



(b) Non-Cross-Domain Targeted Attack

Figure 1. Targeted adversarial attack produces adversarial examples that mislead the deep learning model into predicting a specific class. In cross-domain scenarios, attackers craft adversarial examples with a domain (Domain A) different from the target domain (Domain B), while in non-cross-domain scenarios, attackers and users share the same domain (Domain B).

Fig. 1 shows the targeted adversarial attacks under cross-domain and non-cross-domain attack scenarios. Here, P and Q denote the source domain owned by the attacker and the target domain of the target model, respectively. Fig. 1a describes the scenario of a cross-domain targeted adversarial attack. The attacker crafts targeted adversarial examples using source domain images $x_s \sim P$ on the source model (*i.e*., shadow model), which is then used to mislead the target model to output the predefined target class. Note that there is a special case where the attacker and the target model share the same domain, *i.e*., $P = Q$, as shown in Fig. 1b. We refer to this scenario as the non-cross-domain targeted adversarial attack.

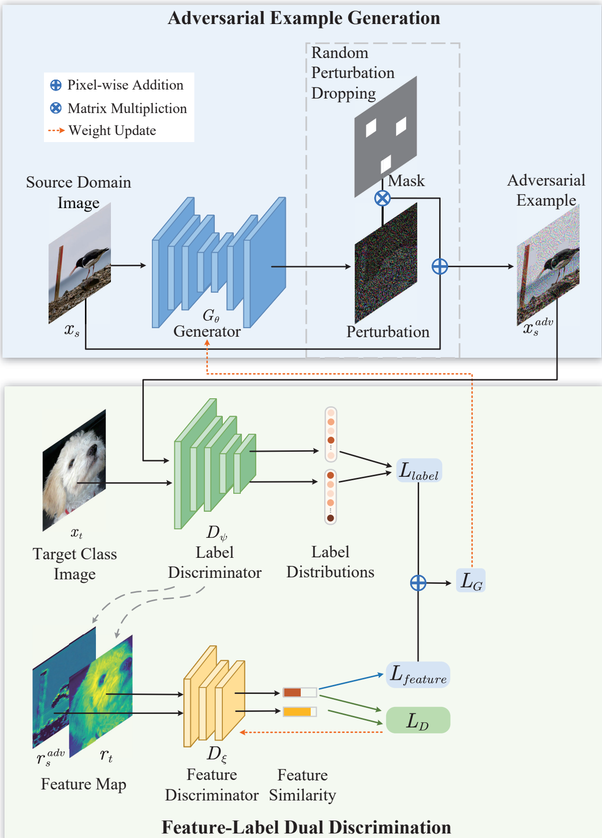


Figure 2. **Framework of TTAA.** The framework comprises two main modules. The adversarial example generation module includes a generator for crafting the adversarial examples and a random perturbation dropping scheme for data augmentation during the training process. The feature-label dual discrimination module distinguishes the adversarial examples and the target class images from feature-wise and label-wise perspectives.

4. Our Method

In this section, we first provide an overview of the proposed TTAA, and then describe the detailed design of each module in our framework and our training strategy.

4.1. Overview of TTAA

Existing transferable adversarial attacks focusing on the cross-model transferability of adversarial examples make a strong assumption that the domain of the target model is accessible for the attackers to perform adversarial attacks in non-cross-domain scenarios, which is not applicable in more practical cross-domain scenarios. Especially for targeted adversarial attacks, the lack of an appropriate description of the target class distribution further limits both the cross-model and cross-domain transferability. Therefore, the key point of the highly transferable targeted adversarial attack is to design a framework that enables the adversarial examples to learn the label-aware and transferable distribution knowledge of the target class images to achieve cross-model and cross-domain transferability.

To address the above issues, we propose TTAA to generate highly transferable targeted adversarial attacks. The basic idea of TTAA is to design an adversarial training method to enable the generated adversarial examples can fully capture the distribution information of the target class from both label and feature perspectives. We present the framework of TTAA in Fig. 2. Specifically, TTAA comprises two main modules, *i.e.*, the adversarial example generation and the label-feature dual discrimination modules. With the input of the image x_s from the source domain, we create the adversarial example x_s^{adv} from the adversarial example generation module, which includes a generator for crafting the perturbation and a random perturbation dropping scheme for data augmentation. Further, the feature-label discrimination module, which consists of the label discriminator and feature discriminator, receives adversarial examples x_s^{adv} and the target class example x_t to make a distinction from both label and feature perspectives. The label discriminator $\mathcal{D}_\psi$ plays two roles: 1) a label classifier for the adversarial example x_s^{adv} and the target class image x_t, capturing label-aware information for both inputs respectively; 2) a feature extractor to create latent feature maps (intermediate layer outputs) for both x_s^{adv} and x_t. The feature discriminator $\mathcal{D}_\xi$ further guides the adversarial examples to learn the feature-aware information of the target class via a feature-wise distance loss. The details of each component and our training strategy are described as follows.

4.2. Label-Feature Dual Discrimination

To help adversarial examples learn the distribution information of the target class from the label perspective, we measure the distance between targeted adversarial examples and target class images. With the input of the image from adversarial example x_s^{adv} and the target class example x_t, the label discriminator outputs the predicted corresponding label $\mathcal{D}_\psi(x_s^{adv})$ and $\mathcal{D}_\psi(x_t)$. Moreover, to further leverage the feature information from x_s^{adv} and x_t, we obtain the corresponding feature map r_s^{adv} and r_t from intermediate layer outputs of $\mathcal{D}_\psi$. To measure the distance between the two label probability distributions, we employ the KL divergence, which can be calculated as follows.

$$
\begin{aligned}
&\mathrm{KL}\Big(\mathcal{D}_\psi\Big(x_s^{adv}\Big)\Big\|\mathcal{D}_\psi\Big(x_t\Big)\Big) = \\
&\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K}\sigma\Big(\mathcal{D}_\psi\Big(x_s^{adv(i)}\Big)\Big)_j \log\frac{\sigma\Big(\mathcal{D}_\psi\Big(x_s^{adv(i)}\Big)\Big)_j}{\sigma\Big(\mathcal{D}_\psi\Big(x_t{}^{(i)}\Big)\Big)_j},
\end{aligned}
\tag{2}
$$

where N denotes the number of input samples, K represents the output dimension of the label discriminator, and $\sigma(\cdot)$ stands for the softmax function with $()_j$ denoting the output probability for label j in the K possible labels. Considering that the KL divergence metric is asymmetric, we further calculate the total label-wise distance (*i.e.*, the label loss) is expressed as follows.

$$
\begin{aligned}
L_{label} = &\mathrm{KL}\left(\mathcal{D}_{\psi}\left(x_s^{adv}\right) \| \mathcal{D}_{\psi}\left(x_t\right)\right) \\
&+ \mathrm{KL}\left(\mathcal{D}_{\psi}\left(x_t\right) \middle\| \mathcal{D}_{\psi}\left(x_s^{adv}\right)\right).
\end{aligned} \tag{3}
$$

Furthermore, to facilitate the adversarial examples to learn the distribution information of the target class from the feature perspective, we measure the similarity between the feature maps r_s^{adv} and r_t extracted by the label discriminator. Specifically, the feature discriminator is a one-class classifier, where the feature maps r_s^{adv} and r_t extracted by the label discriminator serve as inputs and the probabilities of the feature maps r_s^{adv} and r_t being from the target class are outputs, *i.e.*, $\mathcal{D}_{\xi}(r_s^{adv})$ and $\mathcal{D}_{\xi}(r_t)$. The output of 1 from $\mathcal{D}_{\xi}$ indicates that the feature map is from the target class image, and an output of 0 means the opposite. Then, the feature-wise distance between the adversarial example and the target class is computed as

$$
\begin{aligned}
L_{feature} =& \mathrm{BCE}\left(\mathcal{D}_{\xi}\left({r_s}^{adv}\right), 1\right) \\
=& -\sum_{i=1}^{N} \log \mathcal{D}_{\xi}\left(r_s^{adv(i)}\right),
\end{aligned} \tag{4}
$$

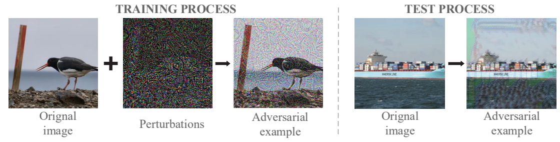
where $\mathrm{BCE}(\cdot)$ denotes the binary cross entropy function.

Combining the label-wise distance L_{label} and the feature-wise distance $L_{feature}$, the distribution information of the target class can be fully captured. With such information, which has high cross-model and cross-domain transferability, we are likely to generate highly transferable targeted adversarial examples via TTAA.

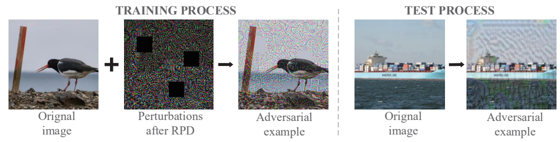
4.3. Random Perturbation Dropping (RPD)

Fig. 3a presents general adversarial example generation from the original image during the training process and test process. Adversarial examples used for training are the simple summations of the original images and perturbations. It is worth noting that there may exist a case where the adversarial perturbations generated by the trained generator are concentrated in a certain region of the image, as shown in the test process of the Fig. 3a, which decreases the robustness of the distribution information of the target class on adversarial examples and compromises the transferability.

Instead, to ensure high transferability, we need to render the perturbations that matter uniformly distributed over the entire image. To achieve this goal, we incorporate the random perturbation dropping (*i.e.*, RPD) technique into the adversarial example generation in the training process. Specifically, as shown in the dashed frame of Fig. 2, the RPD generates a random mask M with the same size as the perturbation, which consists of entries of either 0 or 1. By matrix multiplying the random mask M and the perturbation δ, we can remove several squares of pixels from the perturbation. Formally, after incorporating the random perturbation dropping process, the final generated adversarial image x_s^{adv} is computed as



(a) General Adversarial Example.



(b) Adversarial Example after Random Perturbation Dropping.

Figure 3. Comparison between general adversarial example and adversarial example after random perturbation dropping (RPD) in training and test process. We can observe that by performing RPD on the adversarial examples in the training phase, the generated adversarial examples visually have more density in the testing phase.

$$
x_s^{adv} = M \times \delta + x_s, \tag{5}
$$

where x_s is the original source domain image.

We show the adversarial example generation process with/without the random perturbation dropping in Fig. 3, where Fig. 3a shows the general training process of adversarial examples while Fig. 3b shows the effect of RPD on the adversarial example.

4.4. Generative Adversarial Training

We generalize the adversarial example generation as an optimization problem, for which we design the loss function L_G and L_D for the generator and the feature discriminator, respectively. By minimizing the loss functions, we iteratively update the parameters θ of the generator $\mathcal{G}_{\theta}$ and the parameters ξ of the feature discriminator $\mathcal{D}_{\xi}$ via gradient descent.

$$
\theta \leftarrow \min L_G, \quad \xi \leftarrow \min L_D. \tag{6}
$$

In order for the generator $\mathcal{G}_{\theta}$ to learn the extracted label-feature dual distribution information and generate highly transferable targeted adversarial examples, we design an adversarial training method, which involves the adversarial process between the generator and the feature discriminator. Given the source domain images, x_s, as inputs, the generator $\mathcal{G}_{\theta}$ creates the perturbations δ. To avoid being easily identified by humans, we need to restrict the perturbations to a range of $[-\epsilon, \epsilon]$. Accordingly, the perturbation is generated as follows. We summarize the complete procedures of the generative adversarial training in Algorithm 1.

$$
\delta = \mathrm{clip}\left(\mathcal{G}_{\theta}\left(x_s\right)\right), \tag{7}
$$

where $\mathrm{clip}(\cdot)$ represents a projection operation that restricts the x_s^{adv} to the range of $[x - \epsilon, x + \epsilon]$.

Recall that the training objective of the generator is to successfully deceive the discriminator by minimizing the label-wise and feature-wise distances between the adversarial image and the target class image. Combining Eq. (3) and

Algorithm 1: Training of TTAA

Input: The source domain image x_s, the target class image x_t, pre-trained label discriminator $\mathcal{D}_\psi$, max perturbation ϵ, the number of iteration T, and maximum perturbation magnitude ϵ.

Output: Generator $\mathcal{G}_\theta$

Initialize the generator $\mathcal{D}_\theta$ and feature discriminator $\mathcal{D}_\xi$.

for $i = 1, \cdots, T$ **do**

  Get batches of source domain x_s and the target class image x_t respectively

  Get the adversarial perturbations $\delta = G_\theta(x_s)$

  Generate random mask M for random dropping the perturbations, get adversarial example $x_s^{adv} = M \times \delta + x_s$

  Clip x_s^{adv} to meet $\|x_s^{adv} - x_s\|_\infty \leq \epsilon$

  Get feature maps: r_s^{adv} and r_t

  Calculate label loss:

    $L_{label} = \mathrm{KL}(D_\psi(x_s^{adv}), D_\psi(x_t))$

  Calculate feature loss:

    $L_{feature} = \mathrm{BCE}(D_\xi(r_s^{adv}), 1)$

  Calculate loss of the generator:

    $L_G = L_{label} + L_{feature}$

  Update $\theta \leftarrow \min L_G$

  Calculate loss of feature discriminator:

    $L_D = \mathrm{BCE}(D_\xi(r_s^{adv}), 0) + \mathrm{BCE}(D_\xi(r_t), 1)$

  Update $\xi \leftarrow \min L_D$

end

return *Generator* $\mathcal{D}_\theta$

Eq. (4), the overall loss function of the generator to be minimized is described as

$$L_G = L_{label} + L_{feature}. \tag{8}$$

The training objective of the feature discriminator is to successfully distinguish the feature map of the adversarial image from the target class. In other words, the feature discriminator should output 0 when taking the feature map of the adversarial image as input but output 1 when fed with the feature map of the target class image. The loss function of the feature discriminator $\mathcal{D}_\xi$ is defined as follows.

$$\begin{aligned} L_D &= \mathrm{BCE}\left(\mathcal{D}_\xi\left({r_s}^{adv(i)}\right), 0\right) + \mathrm{BCE}\left(\mathcal{D}_\xi\left({r_t}^{(i)}\right), 1\right) \\ &= -\sum_{i=1}^{N} \log \mathcal{D}_\xi\left(r_s^{adv(i)}\right) - \sum_{i=1}^{N} \log\left(1 - \mathcal{D}_\xi\left({r_t}^{(i)}\right)\right). \end{aligned} \tag{9}$$

5. Performance Evaluation

In this section, we conduct experiments to evaluate the performance of the proposed TTAA. We first present the experimental setup. Then, we sequentially provide the experimental results in non-cross-domain and cross-domain scenarios. We further investigate the performance of TTAAagainst defense methods (*i.e.*, adversarial training). Finally, some ablation studies are performed.

5.1. Experimental Setup

Datasets. Following prior works [17, 21], we use two datasets in our experiments, *i.e.*, the *ImageNet* dataset [31] and the *Paintings* dataset. For the non-cross-domain scenario, we use *ImageNet* as both the source domain and target domain. We set three target classes, *i.e.*, Great Grey Owl, Goose, and French Bulldog, and sample 1300 images for each to form the target class dataset. From the samples of non-target classes, we randomly choose 50,000 images as the training set and 10,000 images as the testing set. For the cross-domain scenario, we take *Paintings* and *ImageNet* as the source domain and target domain, respectively. We follow the same target class dataset generation method as the non-cross-domain scenario. Then, from *Paintings* dataset, we select 50,000 images as the training set and 10,000 images as the testing set. Note that in our experiments, we will report the results averaged over three selected target classes.

Implementation Details. Following baseline methods [16, 17, 21], we choose the ResNet network as the generator. We employ the commonly used models in PyTorch, including VGG-19 [32], ResNet-50 [33], and DenseNet-121 [34], as the label discriminator. Moreover, we use a simple one-class classification model with four convolutional layers as the feature discriminator, which takes the feature maps (*i.e.*, the output of the label discriminator) as input and outputs the probability that the feature map belongs to the target class. Note that each layer of the label discriminator would produce an intermediate result (*i.e.*, feature map), but in our experiments, we use the feature map generated by the 17th layer of the VGG-19 network, the 4th layer of the ResNet-50 network, and the 5th layer of the DenseNet-121 network.

During the training process, the parameters of the label discriminator are fixed, while those of the generator and the feature discriminator are iteratively updated. The training process for each model lasts for 60 epochs with a batch size of 64. We employ the Adam optimizer with an initial learning rate of 0.001. We set the perturbation threshold as $\epsilon = 16/255$.

During the testing process, we use a wide variety of models with different architectures as the target model to evaluate the transferability of adversarial examples, including VGG-16 [32], VGG-19 [32], ResNet-50 [33], ResNet-152 [33], DenseNet-121 [34], DenseNet-201 [34] and SqueezeNet-v1.1 [35]. To further examine the attack performance against possible defenses, we also take some adversarially trained models as the target model, *i.e.*, Adv-VGG-19, Adv-ResNet-50, and Adv-DenseNet-121.

Performance Metric. To measure and compare the effectiveness of targeted adversarial examples, we employ the targeted fooling rate (TFR) as the performance metric, which characterizes the proportion of generated adversarial examples that are predicted by the target model as the attacker-chosen target class. Formally, TFR is computed as

Source \ Target	Attack	VGG-16	VGG-19	Res-50	Res-152	Dense-121	Dense-201	Squeeze-v1.1
VGG-19	PGD	6.3	99.92*	0.28	0.08	0.18	0.14	0.0
	MI-FGSM	13.86	**100.0***	0.68	0.18	0.56	0.62	0.0
	DI-FGSM	22.93	99.64*	1.04	0.24	0.94	0.58	0.0
	GAP	69.71	**100.0***	24.89	27.10	33.56	27.56	1.78
	CDA	71.76	99.14*	25.41	27.96	34.96	26.58	2.44
	TTP	77.25	98.60*	28.76	31.76	41.24	28.14	14.18
	ODI-MI-TI	85.41	99.80*	70.38	43.16	72.75	72.46	14.87
	DRA+PGD	75.02	89.52*	74.04	67.12	87.70	86.24	**74.96**
	ours	**92.95**	99.33*	**79.24**	**67.58**	**95.13**	**89.29**	31.68
Res-50	PGD	0.18	0.22	99.86*	1.0	0.6	0.58	0.0
	MI-FGSM	0.4	0.32	**100.0***	2.31	2.33	4.55	0.0
	DI-FGSM	1.6	1.97	99.84*	11.25	11.34	14.72	0.0
	GAP	49.11	51.78	98.89*	79.56	57.11	52.44	4.22
	CDA	53.75	55.83	99.39*	75.44	70.93	63.24	2.45
	TTP	76.54	62.94	98.73*	78.37	78.64	74.43	35.98
	ODI-MI-TI	70.24	71.79	98.26*	69.34	78.52	88.23	21.60
	DRA+PGD	62.16	59.94	96.54*	**82.18**	84.66	84.89	52.65
	ours	**88.96**	**83.34**	99.16*	80.48	**90.49**	**92.34**	**56.71**
Dense-121	PGD	0.28	0.22	1.32	0.34	99.86*	3.23	0.0
	MI-FGSM	0.54	0.05	2.68	1.12	**100.0***	10.2	0.0
	DI-FGSM	0.82	0.74	5.52	2.57	99.88*	18.34	0.0
	GAP	39.74	42.84	45.4	38.67	99.09*	83.33	3.56
	CDA	43.23	47.22	50.05	34.52	98.94*	87.76	4.61
	TTP	58.31	66.19	62.81	64.54	96.32*	90.02	20.28
	ODI-MI-TI	44.27	58.30	52.79	37.43	99.65*	82.67	19.74
	DRA+PGD	59.56	57.02	79.46	**75.14**	95.92*	83.10	**56.14**
	ours	**72.78**	**72.48**	**81.68**	70.13	99.16*	**91.59**	35.39

Table 1. **Targeted fooling rates (%) of different attacks against different target models in non-cross-domain scenarios**. "*" indicates white-box attack since the target model is the source model, and the best results are highlighted in **bold**.

Source \ Target	Attack	VGG-16	VGG-19	Res-50	Res-152	Dense-121	Dense-201	Squeeze-v1.1
VGG-19	TTP	68.46	-	28.02	30.67	**42.54**	39.62	10.63
	ours	**77.05**	-	**32.63**	**35.20**	39.12	**41.87**	**29.14**
Res-50	TTP	61.85	60.11	-	53.89	74.05	72.14	26.69
	ours	**64.37**	**66.62**	-	**73.44**	**81.02**	**82.38**	**40.67**
Dense-121	TTP	66.75	63.04	51.74	49.72	-	72.43	12.27
	ours	**72.94**	**71.30**	**65.23**	**57.65**	-	**80.42**	**21.71**

Table 2. **Targeted fooling rates (%) of different attacks against different target models in cross-domain scenarios**. The best results are highlighted in **bold**.

$$\text{TFR} = \frac{\sum_{i=1}^{N} g\left(f\left(x_s^{adv(i)}\right), y_t\right)}{N}, g\left(a, b\right) = \begin{cases} 0, a \neq b \\ 1, a = b \end{cases} \quad (10)$$

where N denotes the total number of adversarial examples, $f(\cdot)$ represents the target model, $x_s^{adv(i)}$ stands for the adversarial example generated by the trained generator from the source domain images, and y_t denotes the target class.

Baseline Attacks. We compare TTAA with several state-of-the-art attack methods, including some untargeted adversarial attack methods, *i.e.*, PGD [36], MI-FGSM [10], DI-FGSM [9], GAP [16], CDA [17] and the targeted adversarial attack methods, *i.e.*, TTP [21], ODI [30], DRA [29].

5.2. Results in Non-Cross-Domain Scenarios

We present the experimental results in non-cross-domain scenarios in Tab. 1, where the first column represents the label discriminator (*i.e.*, the source model), while the first row corresponds to the different target models. We can observe that the TFRs of all attack methods are very close to 1 when the target model is the same as the source model. When the target model is different from the source model, we can find that our TTAA leads to larger TFRs than baselines. For example, when the source model is ResNet-50 and the target model is SqueezeNet-v1.1, TTAA achieves a TFR of 56.71%, while the untargeted baseline methods only have TFRs of at most 4.22%. Besides, TTAA outperforms the targeted baseline attack method TTP in terms of 21% higher in TFR. Therefore, TTAA significantly outperforms baselines on the transferability of targeted adversarial examples in the cross-model attack settings.

5.3. Results in Cross-Domain Scenarios

Among all baseline attack methods, only TTP can launch transferable targeted adversarial attacks in cross-domain scenarios. Thus, our experiments in the cross-domain scenarios only compare TTAA to TTP. The TFR results of TTAA and TTP are summarized in Tab. 2. The source models in the first column and the target models in the first row are trained on *Paintings* and *ImageNet*, respectively. Tab. 2 reveals that the proposed TTAA achieves superior performance (*i.e.*, higher TFRs) than TTP. Specifically, when transferring from DenseNet-121 to ResNet-50, TTAA achieves a 13.5% higher TFR than TTP.

What's more, combining Tab. 2 and Tab. 1, it can be observed that TTAA in cross-domain scenarios can achieve even larger TFRs than other methods in non-cross-domain scenarios. For instance, when adversarial examples transfer from DenseNet-121 to ResNet-50, TTAA reached 65.24% targeted fooling rate in Tab. 2 while TTP gets 62.81% targeted fooling rate in Tab. 1. This further validates that TTAA can effectively improve the targeted transferability of adversarial examples, in both non-cross-domain and cross-domain attack scenarios.

5.4. Results against Defenses

We evaluate the attack performance against possible defenses in cross-domain scenarios by setting some adversarially trained models that bear some robustness towards adversarial examples as the target model, and the results are shown in Tab. 3. We can observe that although the overall TFRs of TTAA slightly decrease compared to the non-robust case (*i.e.*, Tab. 2), our method TTAA still outperforms TTP.

Source \ Target	Attack	Adv-VGG-19	Adv-Res-50	Adv-Dense-121
VGG-19	TTP	-	26.97	37.12
	ours	-	**30.75**	**37.45**
Res-50	TTP	56.42	-	72.09
	ours	**63.78**	-	**75.41**
Dense-121	TTP	62.75	45.71	-
	ours	**68.32**	**57.23**	-

Table 3. **Targeted fooling rate of different attacks against adversarially trained target models in cross-domain scenarios**. The best results are highlighted in **bold**.

5.5. Ablation Study

In this subsection, we perform ablation studies to unveil how each of the technical designs (*i.e.*, the label loss, the feature loss, and the random perturbation dropping) affects the performance of TTAA. Fig. 4 shows the TFRs in three cases, where each corresponds to a kind of combination of our technical designs. Note that the case with only label loss design has the lowest transferability (*i.e.*, the lowest TFR). In contrast, when incorporating the feature loss design, the attack performance is improved significantly. This further validates that our feature-label dual discrimination design can greatly help capture the distribution information of the target class. Moreover, we can see that the RPD can further enhance transferability by increasing the diversity of

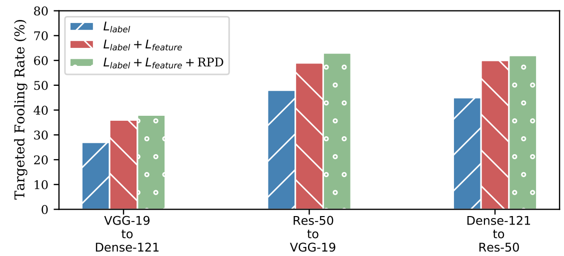


Figure 4. Effect of each component (*i.e.*, label loss L_{label}, feature loss $L_{feature}$, and RPD) of TTAA.

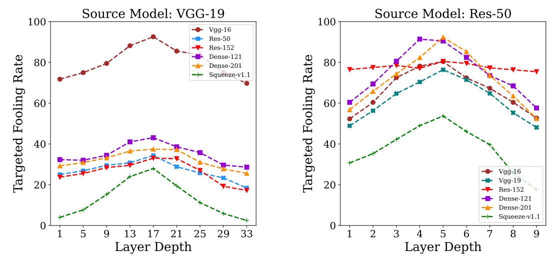


Figure 5. Effect of layer choice of feature maps on TFR. Different layers from the source models (*i.e.*, VGG-19 and Res-50) are selected to extract the feature maps to train the generator.

adversarial examples. Thus, we can conclude that each of our technical designs is effective, and combining them can make a bigger difference.

Fig. 5 displays the TFRs of attacking target models by adversarial examples from generators trained with feature maps of different layer depths. Based on these empirical results, we select the final intermediate layers for each source model as given in Sec. 5.1 to extract the feature maps.

6. Conclusion

In this work, we proposed TTAA to generate highly transferable targeted adversarial examples. Extensive experiments show the superior performance of our attack both in cross-domain and non-cross-domain scenarios compared to those state-of-the-art methods, with an average 7% increase in the fooling rate of the target model. Meanwhile, TTAA can be easily integrated with other targeted adversarial attacks to improve their cross-model and cross-domain targeted transferability. Moreover, the experiments also demonstrate our attack can evade defense of adversarial training with only about a 3% drop in the fooling rate of the target model, indicating the robustness of our attack.

Acknowledgements

This work was supported by Key R&D Program of Zhejiang (Grant No. 2022C01018), National Natural Science Foundation of China (Grants No. 62122066, U20A20182, 61872274, 62102337) and National Key R&D Program of China (Grant No. 2021ZD0112803).

References

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 1

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016. 1

[3] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. Ieee, 2013. 1

[4] William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4960–4964. IEEE, 2016. 1

[5] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deeply learned face representations are sparse, selective, and robust. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2892–2900, 2015. 1

[6] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708, 2014. 1

[7] Ekim Yurtsever, Jacob Lambert, Alexander Carballo, and Kazuya Takeda. A survey of autonomous driving: Common practices and emerging technologies. *IEEE access*, 8:58443–58469, 2020. 1

[8] 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. 1, 2

[9] 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*, pages 2730–2739, 2019. 1, 3, 7

[10] 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*, pages 9185–9193, 2018. 1, 2, 7

[11] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. *arXiv preprint arXiv:1611.02770*, 2016. 1, 2

[12] Dongxian Wu, Yisen Wang, Shu-Tao Xia, James Bailey, and Xingjun Ma. Skip connections matter: On the transferability of adversarial examples generated with resnets. *arXiv preprint arXiv:2002.05990*, 2020. 1, 2

[13] Nathan Inkawhich, Wei Wen, Hai Helen Li, and Yiran Chen. Feature space perturbations yield more transferable adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7066–7074, 2019. 1, 2

[14] Zhibo Wang, Hengchang Guo, Zhifei Zhang, Wenxin Liu, Zhan Qin, and Kui Ren. Feature importance-aware transferable adversarial attacks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7639–7648, 2021. 1, 3

[15] Lianli Gao, Yaya Cheng, Qilong Zhang, Xing Xu, and Jingkuan Song. Feature space targeted attacks by statistic alignment. *arXiv preprint arXiv:2105.11645*, 2021. 1, 2, 3

[16] Omid Poursaeed, Isay Katsman, Bicheng Gao, and Serge Belongie. Generative adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4422–4431, 2018. 1, 3, 6, 7

[17] 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, 2019. 1, 3, 6, 7

[18] Qilong Zhang, Xiaodan Li, Yuefeng Chen, Jingkuan Song, Lianli Gao, Yuan He, and Hui Xue. Beyond imagenet attack: Towards crafting adversarial examples for black-box domains. *arXiv preprint arXiv:2201.11528*, 2022. 1, 3

[19] Nathan Inkawhich, Kevin J Liang, Lawrence Carin, and Yiran Chen. Transferable perturbations of deep feature distributions. *arXiv preprint arXiv:2004.12519*, 2020. 2, 3

[20] Nathan Inkawhich, Kevin Liang, Binghui Wang, Matthew Inkawhich, Lawrence Carin, and Yiran Chen. Perturbing across the feature hierarchy to improve standard and strict blackbox attack transferability. *Advances in Neural Information Processing Systems*, 33:20791–20801, 2020. 2, 3

[21] Muzammal Naseer, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Fatih Porikli. On generating transferable targeted perturbations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7708–7717, 2021. 2, 3, 6, 7

[22] Aditya Ganeshan, Vivek BS, and R Venkatesh Babu. Fda: Feature disruptive attack. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8069–8079, 2019. 2

[23] Mathieu Salzmann et al. Learning transferable adversarial perturbations. *Advances in Neural Information Processing Systems*, 34:13950–13962, 2021. 2

[24] 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. 2

[25] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. Ieee, 2017. 2

[26] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1625–1634, 2018. 2

[27] Lei Wu, Zhanxing Zhu, Cheng Tai, et al. Understanding and enhancing the transferability of adversarial examples. *arXiv preprint arXiv:1802.09707*, 2018. 2

[28] Zhibo Wang, Mengkai Song, Siyan Zheng, Zhifei Zhang, Yang Song, and Qian Wang. Invisible adversarial attack against deep neural networks: An adaptive penalization approach. *IEEE Transactions on Dependable and Secure Computing*, 18(3):1474–1488, 2019. 2

[29] Yao Zhu, Yuefeng Chen, Xiaodan Li, Kejiang Chen, Yuan He, Xiang Tian, Bolun Zheng, Yaowu Chen, and Qingming Huang. Toward understanding and boosting adversarial transferability from a distribution perspective. *IEEE Transactions on Image Processing*, 31:6487–6501, 2022. 3, 7

[30] Junyoung Byun, Seungju Cho, Myung-Joon Kwon, Hee-Seon Kim, and Changick Kim. Improving the transferability of targeted adversarial examples through object-based diverse input. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15244–15253, 2022. 3, 7

[31] Alexey Kurakin, Ian Goodfellow, Samy Bengio, Yinpeng Dong, Fangzhou Liao, Ming Liang, Tianyu Pang, Jun Zhu, Xiaolin Hu, Cihang Xie, et al. Adversarial attacks and defences competition. In *The NIPS'17 Competition: Building Intelligent Systems*, pages 195–231. Springer, 2018. 6

[32] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 6

[33] 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*, pages 770–778, 2016. 6

[34] 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*, pages 4700–4708, 2017. 6

[35] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016. 6

[36] 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. 7