Improving the Transferability of Targeted Adversarial Examples through Object-Based Diverse Input

Junyoung Byun Seungju Cho Myung-Joon Kwon Hee-Seon Kim Changick Kim
Korea Advanced Institute of Science and Technology (KAIST)
{bjyoung, joyga, kwon19, hskim98, changick}@kaist.ac.kr

Abstract

The transferability of adversarial examples allows the deception on black-box models, and transfer-based targeted attacks have attracted a lot of interest due to their practical applicability. To maximize the transfer success rate, adversarial examples should avoid overfitting to the source model, and image augmentation is one of the primary approaches for this. However, prior works utilize simple image transformations such as resizing, which limits input diversity. To tackle this limitation, we propose the object-based diverse input (ODI) method that draws an adversarial image on a 3D object and induces the rendered image to be classified as the target class. Our motivation comes from the humans' superior perception of an image printed on a 3D object. If the image is clear enough, humans can recognize the image content in a variety of viewing conditions. Likewise, if an adversarial example looks like the target class to the model, the model should also classify the rendered image of the 3D object as the target class. The ODI method effectively diversifies the input by leveraging an ensemble of multiple source objects and randomizing viewing conditions. In our experimental results on the ImageNet-Compatible dataset, this method boosts the average targeted attack success rate from 28.3% to 47.0% compared to the state-of-the-art methods. We also demonstrate the applicability of the ODI method to adversarial examples on the face verification task and its superior performance improvement. Our code is available at https://github.com/dreamflake/ODI.

1. Introduction

Deep learning models have demonstrated outstanding performance in a variety of fields and have permeated our daily lives [6, 10, 13]. However, adversarial examples show that these models are vulnerable to maliciously crafted small input perturbations [7, 35]. Interestingly, an adversarial example that is generated to attack a network is likely to disturb other networks as well. This intriguing property of adversarial examples is known as *transferability* [21, 26, 37]. This property allows an adversary to attack a black-box target model without knowing its interior.

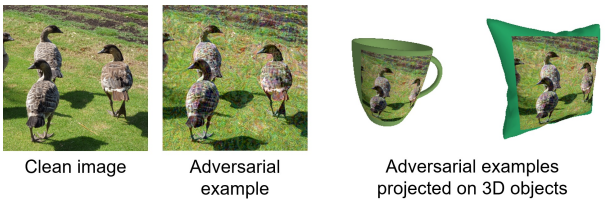


Figure 1. **Illustrations of our motivation.** If a targeted adversarial example really looks like the target class to the model, the model should also classify the adversarial examples projected on the 3D objects as the target class.

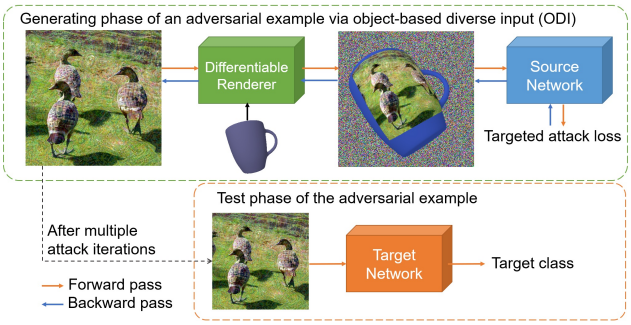


Figure 2. **The framework of targeted adversarial attacks with the proposed object-based diverse input (ODI) method.** Please note that the ODI method exploits 3D adversarial objects in the generating phase only. It finally improves the transferability of 2D adversarial examples.

On black-box models, targeted attacks are significantly more challenging compared to non-targeted attacks which simply induce the victim models to malfunction without specifying a target class [21, 43]. Targeted attacks demand further exploration since they can cause more serious problems by deceiving models into predicting a designated harmful target class. Research on these transfer-based targeted attacks is important since it can help service providers prepare their models against these potential threats and assess the robustness of their models.

The transfer success rates greatly vary depending on the difference between the source and target models. Various approaches have been presented to improve the transferability, such as introducing momentum [7, 22] and different loss functions [21, 43] for better optimization, input data augmentation [41, 44], and utilizing an ensemble of multiple source models [23].

Among these strategies, we focus on input transformation-based methods and their limitations. These methods create adversarial examples that are robust against image transforms such as random resizing [41, 44] and translation [8] to prevent overfitting to the source model. However, since these methods use simple image transformations, they limit the diversity of input.

Our motivation for tackling this limitation comes from the humans' superior perception of an image printed on a 3D object (*e.g.*, promotional merchandise commonly distributed at event booths). As a 2D image is projected on a 3D object, the original image is bent, the color looks different due to illumination, and some parts of the image are invisible depending on the viewpoint. Nevertheless, if the image is clear enough, humans can recognize the image content on the 3D object in a variety of viewing conditions. Likewise, if an adversarial example really looks like the target class to the source model, the model should also recognize the target class in the image printed on 3D objects. Our motivation is illustrated in Fig. 1.

From this motivation, we propose the **object-based diverse input (ODI)** method for boosting the transferability of targeted adversarial examples. Specifically, we introduce 3D objects and project an adversarial example on the objects' surfaces. Then, we induce the rendered objects to be classified as the target class in a variety of rendering environments, including different lighting and viewpoints. This realistic input diversification can generalize the attack ability and improve the transferability of the adversarial example. The overall scheme is illustrated in Fig. 2.

Our contributions can be listed as follows.

- We propose the object-based diverse input (ODI) method to enhance the transferability of targeted adversarial examples. To our knowledge, this is the first time that 3D objects are used as canvases for 2D adversarial examples during their optimizations.

- We discovered that the attack success rate varies depending on the 3D object (*e.g.*, a pillow and a cup). Our experimental results also indicate that an ensemble of carefully chosen source objects can further improve transferability.

- In the experimental results with four source models and ten target models on the ImageNet dataset, the proposed ODI method boosts the average targeted attack success rate from 28.3% to 47.0% compared to the combination of state-of-the-art methods.

- We also demonstrate the applicability of the ODI method to adversarial examples on the face verification task and its overwhelming performance improvement.

2. Related Work

2.1. Adversarial Attacks in the Black-Box Setting

In the black-box setting, since adversaries cannot access the interiors of the target model, the gradient of the image cannot be directly calculated using backpropagation. Query-based attacks [3–5] use multiple queries to find an adversarial example via gradient estimation [4,5] or random search [1, 3]. However, they are based on the unreasonable assumption that the output of the target model can be obtained via queries.

In comparison, transfer-based attacks [21, 24, 38] can generate adversarial examples that deceive the target model without requiring a query. Specifically, transfer-based attack methods generate adversarial examples through a white-box attack on a surrogate model, and the attacker expects the transferability of the adversarial example and attempts to deceive the target model with the generated image [26, 27]. Therefore, we need to generate highly transferable adversarial examples that are capable of deceiving unknown models using a white-box surrogate model.

Adversarial attacks in the white-box scenario take advantage of the gradient of the loss function with respect to the image. The standard algorithm for ℓ_∞-norm-constrained adversarial perturbations utilizes the sign of the gradient, which is called the fast gradient sign method (FGSM) [11]. Formally, let f be the classifier and $\mathcal{L}$ be the loss function for targeted attacks. Then, the targeted adversarial example $\mathbf{x}_{adv}$ can be found by solving the following optimization problem, given an image $\mathbf{x}$ and a target label y_t.

$$\mathbf{x}_{adv} = \mathbf{x} - \epsilon \cdot \text{sign}(\nabla_{\mathbf{x}} \mathcal{L}(f(\mathbf{x}), y_t)), \tag{1}$$

where ϵ represents the step size. It updates the image $\mathbf{x}$ to minimize the loss for targeted attacks. It can be further optimized by iterating updates $\mathbf{x}$ on Eq. (1) with a smaller step size α, and this iterative version is called iterative-FGSM (I-FGSM) [19]. To aid in avoiding local minima to improve transferability, Dong *et al.* [7] incorporate a momentum term in the optimization, which is referred to as momentum iterative FGSM (MI-FGSM).

In addition to these fundamental adversarial attacks, various techniques have been proposed to improve transferability by helping the image avoid falling into local minima and prevent overfitting to a specific source model.

One common approach is input diversification. The diverse inputs (DI) method [41] applies random resizing and

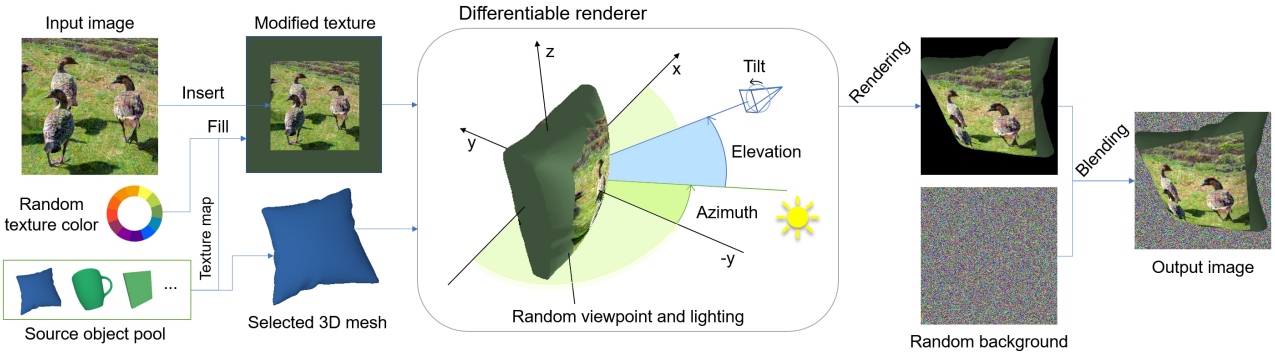


Figure 3. The pipeline of the object-based diverse input (ODI) method.

padding to the image with probability p for each inference in the iterative attacks to minimize overfitting to the source model. The resized-diverse-inputs (RDI) method [44] is similar to the DI method, but it resizes the expanded and padded image back to its original size at the final step of the DI. Unlike DI, which stochastically applies the image transform, RDI always applies the resizing image transform (*i.e.*, $p = 1$).

The translation-invariant (TI) attack method [8] computes a weighted average of gradients from a set of translated images within a specified range, giving a higher weight to smaller displacements. To minimize computing time, TI approximates the weighted mean of the gradients by applying Gaussian blur to the original gradient. Updating the image with the obtained gradient mitigates the adversarial example's overfitting to the source model.

Wu *et al*. [39] highlight the limitations of heuristic image transformations like resizing [41] and propose the adversarial transformation-enhanced transfer attack (ATTA). They train an adversarial transformation network that neutralizes adversarial examples within an adversarial learning framework. Then, they construct a more robust adversarial example that is resistant to the trained adversarial transform. However, since their adversarial transformation network is based on a 2-layer CNN, the network can perform only simple image transformations, such as blurring and sharpening.

The scale-invariant (SI) attack method [22] generates several scale variants of an image by altering the scale of pixel values and computing the gradient from them for each iteration. This promotes the transferability of adversarial examples by minimizing overfitting to the source model.

The recently proposed variance tuning (VT) method [37] focuses on gradient variance, defined as the difference between an image's gradient and the average gradients of nearby images. By reducing the gradient variance, this can stabilize the update direction.

On the other hand, various approaches have been presented to improve transferability by using different loss functions for targeted attacks. Li *et al*. [21] identify the issue of the widely used cross-entropy loss, which results in vanishing gradient problems in iterative targeted attacks. To address this issue and increase the transferability, they adapt the size of the gradients using the Poincare distance. Zhao *et al*. [43] point out that prior work uses an inappropriately small number of iterations in the optimization of targeted adversarial examples. They emphasize that the following simple logit loss $\mathcal{L}_{logit}$ for targeted attacks can achieve state-of-the-art performance with sufficient iterations.

$$\mathcal{L}_{logit}(f(\mathbf{x}^{adv}), y_t) = -\ell_t(f(\mathbf{x}^{adv})), \tag{2}$$

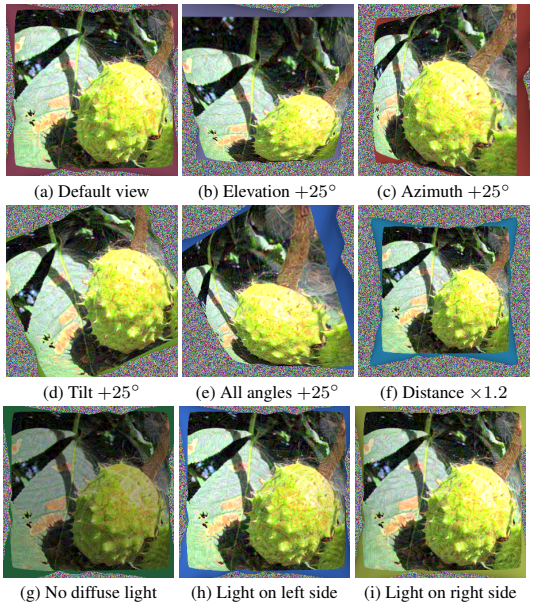
where ℓ_t is the logit output corresponding to the target class.

2.2. Adversarial Attacks with Differentiable Rendering

Differentiable rendering projects 3D objects onto 2D images, and by making the internal process differentiable, it allows computing the gradient of the 3D objects' attributes, such as mesh and texture [17]. This differentiable rendering enables the optimization of 3D objects via digital simulation, and is commonly used to generate physically applicable adversarial examples that are robust under various viewpoints [2, 40, 42]. The most significant difference between these methods and the ODI method is that the prior works treat an adversarial mesh as their **goal**, but the ODI method treats it as a **tool** for improving the transferability of a 2D adversarial image.

3. Methodology

The object-based diverse input method preprocesses images before feeding them to the source network during the iterative optimization of adversarial examples. Since the ODI method uses a differentiable renderer, even if the rendered image is fed into the network, the gradient of the adversary's objective loss with respect to the input image can be computed via back-propagation. The overall pipeline of the ODI method is illustrated in Fig. 3, and the detailed algorithm of the ODI method is described in the supplemen-



(a) Default view (b) Elevation +25° (c) Azimuth +25°
(d) Tilt +25° (e) All angles +25° (f) Distance ×1.2
(g) No diffuse light (h) Light on left side (i) Light on right side

Figure 4. Rendered images with different rendering parameters.

tary material. The entire procedure of the ODI method can be broken down into three stages, which we will cover in detail.

Preparation of an adversarial 3D mesh. The ODI method employs a 3D object as a canvas to draw an adversarial example. Thus, it does not matter if the object changes during iterations. This is where the ODI method significantly differs from previous studies [2,40,42] that employ 3D meshes for generating physical adversarial examples. First, we randomly select an object from the source object pool. The selected object has a triangular mesh, a texture map, and a bounding box that indicates the canvas region on which the adversarial example will be drawn in the texture map. Next, we fill the texture map with a random solid color, and then we resize and insert the adversarial example into the texture map's bounding box region. Within the frame area, we can also leverage existing input diversification techniques, such as RDI, but we exclude them to clearly demonstrate the effectiveness of ODI in comparison to existing approaches.

Rendering environment setup. The rendering environment includes lighting and cameras, which are required to render 3D objects. For the camera, we fix the intrinsic parameters and adjust the extrinsic parameters. We alter the three camera angles: elevation, azimuth, and tilt. Please refer to Fig. 3 for the definition of each camera angle, and Fig. 4 for demonstrations of how each angle alters the viewpoint. In the ODI method, the 3D mesh is initially scaled so that the projected image occupies about 85% of the rendered image in the default view. The three camera angles and camera distance are randomly sampled within a preset range.

There are two primary types of lights for illumination models — directional lights and point lights. We employ

Algorithm 1 ODI-MI-TI-FGSM

Input: A clean example $\mathbf{x}$; a target label y_t; a classifier f.
Input: Adversary's loss function $\mathcal{L}$; ℓ_∞ perturbation constraint ϵ; step size α; maximum iterations T; decay factor μ; and Gaussian kernel $\boldsymbol{W}$.
Output: An adversarial example $\mathbf{x}^{adv}$

$\mathbf{g}_0 = 0$; $\mathbf{x}_0^{adv} = \mathbf{x}$
for $t = 0 \rightarrow T-1$ **do**
  Calculate the gradient $\hat{\mathbf{g}}_{t+1}$ ▷ Apply ODI

$$\hat{\mathbf{g}}_{t+1} = \nabla_{\mathbf{x}_t^{adv}} \mathcal{L}(f(ODI(\mathbf{x}_t^{adv})), y_t) \quad (3)$$

  $\tilde{\mathbf{g}}_{t+1} = \mu \cdot \mathbf{g}_t + \frac{\hat{\mathbf{g}}_{t+1}}{\|\hat{\mathbf{g}}_{t+1}\|_1}$ ▷ Apply MI
  $\mathbf{g}_{t+1} = \boldsymbol{W} * \tilde{\mathbf{g}}_{t+1}$ ▷ Apply TI
  $\mathbf{x}_{t+1}^{adv} = \mathbf{x}_t^{adv} - \alpha \cdot \text{sign}(\mathbf{g}_{t+1})$ ▷ Apply FGSM
  $\mathbf{x}_{t+1}^{adv} = Clip_{\mathbf{x}}^{\epsilon}(\mathbf{x}_{t+1}^{adv})$
end for
$\mathbf{x}^{adv} = \mathbf{x}_T^{adv}$
return $\mathbf{x}^{adv}$

point lights in our work, but directional lights can also be used. We randomly adjust the brightness of ambient and diffuse light within a preset range. Additionally, we alter the position of the light by adding a random displacement to its base position, causing it to be randomly placed within a box. Figure 4 illustrates examples of rendered images with different lighting.

Rendering and blending with backgrounds. Finally, we render the adversarial 3D mesh in the sampled environment and blend it with a randomly generated background image to create the final output image. This image will be fed into the source network, which *aids in optimizing the input image to appear as the target class in a wide variety of contexts, enhancing transferability*. We present the algorithm of the ODI-MI-TI-FGSM method in Algorithm 1.

4. Experiments and Discussion

4.1. Experimental Settings

Dataset and general settings. We utilized the DEV set of the ImageNet-Compatible dataset[1], which has been widely used in previous works [21, 43]. This dataset provides 1,000 299×299-sized images with their target classes. We adopted the widely used ℓ_∞-norm perturbation constraint $\epsilon = 16/255$. Following [43], we used the step size $\alpha = 2/255$ for the iterative attacks. Our approach and all baselines leveraged the simple logit loss (Eq. (2)) proposed in

[1] https://github.com/cleverhans-lab/cleverhans/tree/master/cleverhans_v3.1.0/examples/nips17_adversarial_competition/dataset

Source : RN-50	Target model									
Attack	VGG-16	RN-18	DN-121	Inc-v3	Inc-v4	Mob-v2	IR-v2	Adv-Inc-v3	Ens-adv-IR-v2	Computation time per image (sec)
DI-MI-TI	62.2	54.9	71.4	10.5	9.0	28.5	4.5	0.0	0.0	2.7
RDI-MI-TI	67.8	73.4	82.9	32.4	24.6	44.5	17.4	0.0	0.0	2.3
RDI-MI-TI-SI	71.2	81.7	88.5	56.6	42.8	58.0	36.6	0.2	0.9	11.2
RDI-MI-TI-VT	70.3	78.7	82.5	44.6	39.1	54.4	33.7	0.2	1.7	14.1
ODI-MI-TI	76.8	77.0	86.8	67.4	55.4	66.8	48.0	0.7	1.7	6.0
ODI-MI-TI-VT	**81.6**	**84.4**	**89.2**	**74.5**	**65.9**	**75.6**	**62.3**	**4.6**	**8.7**	57.2
Source : VGG-16	Target model									
Attack	RN-18	RN-50	DN-121	Inc-v3	Inc-v4	Mob-v2	IR-v2	Adv-Inc-v3	Ens-adv-IR-v2	Computation time per image (sec)
DI-MI-TI	7.6	11.2	12.7	0.6	2.3	6.3	0.2	0.0	0.0	6.1
RDI-MI-TI	28.7	31.5	35.9	6.6	9.5	18.0	3.7	0.0	0.0	5.4
RDI-MI-TI-SI	48.0	45.6	55.2	20.1	21.0	28.4	9.9	0.0	0.0	26.2
RDI-MI-TI-VT	42.5	35.5	44.3	13.4	19.0	23.4	8.3	0.0	0.0	31.8
ODI-MI-TI	60.8	64.3	71.1	37.0	38.0	47.0	21.1	0.0	0.0	9.0
ODI-MI-TI-VT	**72.3**	**72.0**	**76.6**	**48.7**	**47.9**	**57.6**	**34.7**	**0.4**	**0.7**	71.9
Source : DN-121	Target model									
Attack	VGG-16	RN-18	RN-50	Inc-v3	Inc-v4	Mob-v2	IR-v2	Adv-Inc-v3	Ens-adv-IR-v2	Computation time per image (sec)
DI-MI-TI	38.3	30.1	43.6	7.1	7.7	13.7	4.5	0.0	0.0	2.8
RDI-MI-TI	41.9	45.3	55.9	21.0	19.1	21.8	12.9	0.0	0.0	2.5
RDI-MI-TI-SI	44.2	54.5	59.7	34.7	24.9	26.9	22.6	0.2	0.5	12.1
RDI-MI-TI-VT	49.1	56.1	63.3	32.1	28.8	29.4	25.6	0.3	0.8	15.2
ODI-MI-TI	64.5	63.4	71.6	53.5	46.4	44.2	38.3	0.4	0.7	6.2
ODI-MI-TI-VT	**70.7**	**74.6**	**79.1**	**64.1**	**57.5**	**57.8**	**53.0**	**2.3**	**5.0**	71.7
Source : Inc-v3	Target model									
Attack	VGG-16	RN-18	RN-50	DN-121	Inc-v4	Mob-v2	IR-v2	Adv-Inc-v3	Ens-adv-IR-v2	Computation time per image (sec)
DI-MI-TI	4.2	2.2	3.6	5.4	4.3	2.4	3.6	0.0	0.0	2.2
RDI-MI-TI	3.2	4.4	4.4	6.9	8.3	3.0	5.5	0.0	0.0	1.9
RDI-MI-TI-SI	4.2	7.2	6.3	10.3	10.5	5.0	11.4	0.2	0.3	9.2
RDI-MI-TI-VT	4.8	8.5	8.9	11.9	14.6	6.2	12.8	0.2	0.0	11.8
ODI-MI-TI	15.7	14.7	17.4	30.4	32.1	14.1	26.9	0.3	0.6	5.5
ODI-MI-TI-VT	**26.7**	**29.1**	**34.0**	**52.5**	**50.8**	**25.4**	**45.8**	**1.9**	**3.3**	62.6

Table 1. Targeted attack success rates (%) against nine black-box target models with the four source models. For each attack, we also reported the average computation time to generate an adversarial example.

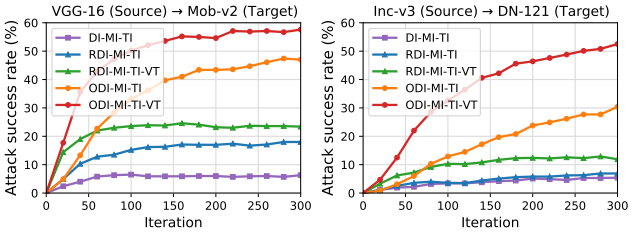


Figure 5. Targeted attack success rates (%) according to the number of iterations.

[43] which is superior in targeted attacks. Following [43], each iterative attack method runs for 300 iterations (*i.e.*, $T = 300$). Each iterative attack was performed using a single NVIDIA RTX 2080Ti GPU.

Source and target models. For fair comparison with existing works, we adopted four models used in [43] as source and target models in our experiments — ResNet-50 (RN-50) [13], Inception-v3 (Inc-v3) [34], DenseNet-121 (DN-121) [14], and VGG-16_bn (VGG-16) [32]. Additionally, we added six additional models in the collection of target models for more comprehensive comparisons — ResNet-18 (RN-18) [13], Inception-v4 (Inc-v4) [33], MobileNet-v2 (Mob-v2) [30], Inception ResNet-v2 (IR-v2) [33], adversarially trained Inc-v3 (Adv-Inc-v3) [20], and ensemble-adversarially trained IR-v2 (Ens-adv-IR-v2) [20]. This paper focuses on single-source model-based transfer attacks to demonstrate the ODI method's effectiveness in a challenging environment. However, we believe that the ensemble of source models can further boost the transfer success rates.

Baselines. We employed four baseline attack methods,

Source : DN-121	Target model								
Source object pool	VGG-16	RN-18	RN-50	Inc-v3	Inc-v4	Mob-v2	IR-v2	Adv-Inc-v3	Ens-adv-IR-v2
{*Package*}	59.8	54.8	65.6	43.5	37.6	35.8	29.8	0.1	0.5
{*Cup*}	34.2	45.7	47.9	37.8	29.8	28.5	26.7	**0.5**	**1.2**
{*Pillow*}	64.1	57.6	68.6	44.9	40.6	39.4	30.5	0.1	0.6
{*T-shirt*}	23.5	36.4	38.1	31.2	23.5	19.3	19.4	**0.5**	1.0
{*Ball*}	46.1	26.3	36.7	17.1	16.6	17.7	10.6	0.0	0.0
{*Book*}	50.9	61.9	67.0	52.8	39.5	39.5	36.6	0.3	1.0
{All 6 objects}	60.2	59.5	66.3	48.8	42.7	42.4	35.7	0.4	0.9
{*Package, Pillow, Book*}	**64.5**	**63.4**	**71.6**	**53.5**	**46.4**	**44.2**	**38.3**	0.4	0.7

Table 2. Targeted attack success rates (%) of ODI-MI-TI against nine black-box target models with different source object pools. The ensembles of multiple source objects outperform their single object counterparts.

(a) *Package* (b) *Cup* (c) *Pillow*

(d) *T-shirt* (e) *Ball* (f) *Book*

Figure 6. Six source objects used in our experiments. An image is printed on them to visualize the area of adversarial texture.

which are various combinations of six existing techniques: DI [41], RDI [44], MI [7], TI [8], SI [22], and VT [37]. Please note that the previously reported state-of-the-art method is DI-MI-TI with simple logit loss [43]. However, we further improved this method by replacing DI with RDI, combining it with the recently proposed VT and SI, and using them as the state-of-the-art baselines (RDI-MI-TI-SI and RDI-MI-TI-VT). The maximally enlarged image sizes of DI and RDI were set to 330×330 and 340×340, respectively. Following [43], the convolution kernel size for TI was set to 5 and p for DI to 0.7, while the decay factor μ for MI was set to 1.0. The numbers of scales and samples for SI and VT were set to 5, and β for VT to 1.5.

We also implemented ATTA-MI-TI, but ATTA [39] was initially designed for non-targeted attacks with few iterations, so it performed poorly. Although we changed ATTA to use targeted adversarial examples using logit loss for training, its results were not comparable. We hypothesize that the repeated use of the fixed transformation limits transferability. For a rich comparison, we included the results of MI-TI and ATTA-MI-TI in supplementary material.

Settings for the ODI method. We utilized the PyTorch3D library [28] for the differentiable rendering in the ODI method. For the parameters of the ODI method, we constructed the source object pool as {*Package, Pillow, Book*}, each of which is shown in Fig. 6. The ranges of the solid texture color, camera angles and distance were set to $[0.1, 0.7]$, $[-35°, 35°]$, and $[0.8, 1.2]$, respectively. The default position of the light was set to $[0, 0, 4]$, and its maximum displacement to 2. The brightness range of ambient light was set to $[0.6, 0.9]$ and the brightness range of diffuse light to $[0, 0.5]$. Finally, we set the shininess connected to the material's reflectance to 0.5.

4.2. Experimental Results

Transfer success rates. Table 1 shows the targeted attack success rates against nine black-box target models with the four source models. The ODI-MI-TI-VT enhanced the average attack success rate from 28.3% to 47.0% compared to the best performance of baselines. The ODI method's performance improvements were most prominent when the source model was VGG-16. Compared to DI-MI-TI, which is the previous state-of-the-art technique, ODI-MI-TI and ODI-MI-TI-VT boosted the average transfer success rates from 4.5% to 37.7% (8×) and 45.7% (10×), respectively. The targeted attack success rates for two cases are shown in Fig. 5, and the rest of the plots can be found in supplementary material.

Computation time. In Table 1, we also reported the average time required to generate an adversarial example. Due to the rendering overhead, the ODI method requires more computational cost than DI and RDI. However, when compared to RDI-MI-TI-VT and RDI-MI-TI-SI, ODI-MI-TI cut the required time by half and increased the attack success rate by 10.8% on average. When VT or SI is used, the amount of computing required increases significantly because each loop requires more inference in proportion to the numbers of scales and samples.

Attacks on adversarially trained models. Adversarial

Source : Inc-v3		Target model								
Ablation	Value	VGG-16	RN-18	RN-50	DN-121	Inc-v4	Mob-v2	IR-v2	Adv-Inc-v3	Ens-adv-IR-v2
Angle	$-5° \sim 5°$	5.4	4.1	4.8	8.1	7.8	3.7	7.0	0.2	0.0
	$-15° \sim 15°$	9.1	7.7	10.9	19.4	20.7	7.4	16.7	0.0	0.1
	$-25° \sim 25°$	13.4	11.9	14.2	28.7	28.6	11.3	22.7	0.2	0.0
	$-35° \sim 35°$	**15.6**	**14.1**	**17.9**	**31.6**	**30.5**	12.9	**24.9**	**0.3**	**0.4**
	$-45° \sim 45°$	15.4	**14.1**	16.9	30.1	26.0	**13.2**	22.2	0.2	0.1
Distance	$0.8\times \sim 1.2\times$	**15.6**	**14.1**	**17.9**	**31.6**	**30.5**	**12.9**	**24.9**	**0.3**	**0.4**
	$0.9\times \sim 1.1\times$	12.5	12.4	14.6	26.9	26.0	11.2	21.2	0.1	0.2
	$1.0\times$	11.1	11.4	12.6	23.4	23.4	9.1	18.3	0.0	0.1
Background	Random pixel	15.6	**14.1**	**17.9**	**31.6**	**30.5**	12.9	**24.9**	**0.3**	**0.4**
	Random solid	15.7	13.4	**16.9**	30.7	27.9	12.5	22.3	0.2	0.3
	Blurred image	**15.8**	12.5	17.0	**31.6**	28.5	12.3	22.1	**0.3**	0.3
	Black solid	14.7	13.1	16.2	28.1	28.4	**13.1**	21.9	**0.3**	0.1

Table 3. Targeted attack success rates (%) against nine black-box target models with different camera parameters and backgrounds. For these experiments, we used *Pillow* as the source object.

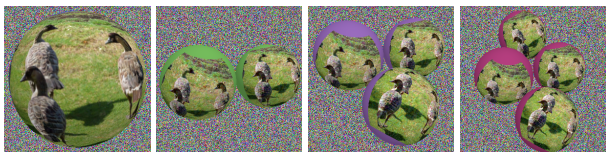
training is widely considered to be one of the most effective adversarial defenses. Targeted attacks on adversarially trained models [25, 36] are more challenging in the black-box setting. Hence, the previously reported state-of-the-art method, DI-MI-TI, recorded 0% attack success rate on Adv-Inc-v3 [20, 25] and Ens-adv-IR-v2 [20, 36]. Surprisingly, ODI-MI-TI-VT showed up to 4.6% and 8.7% attack success rates against these models. Given the significant architectural differences between the source and target models, this improvement is noteworthy.

4.3. Ablation Studies

In this section, we conduct extensive ablation studies on the proposed approach and describe our findings.

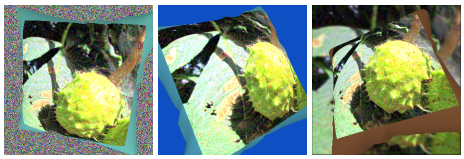
Variation of source object pools. Since the 3D source object in the ODI method is like a canvas for adversarial examples, the object can be freely selected from various 3D objects. As candidate source models, we chose six objects[2] that are commonly observed in our daily life, as shown in Fig. 6. Based on these models, we performed single-object-based and multi-object-based attacks. The results are summarized in Table 2. Interestingly, among single-object-based attacks, *Pillow* had the highest attack success rate against VGG-16, whereas *Book* recorded the highest success rate against Inc-v3. This means that an object's transferability enhancement varies depending on the target model. Moreover, the results of multi-object-based attacks reveal that the transferability enhancements of each 3D object complement each other. Finally, an ensemble of three objects outperformed the ensemble of all objects, highlighting the importance of carefully assembling the source object pool.

Variation of the number of objects. Instead of rendering a single object, we can visualize multiple objects to diversify the input. Using the balls as shown in Fig. 7, we study the transferability of the adversarial example as a function of the number of objects. The attack success rate for each target model can vary based on the number of source objects, even though the same 3D object is used. When the source model was VGG-16, *Three balls* showed the highest success rates for most cases, but adversarially trained models were most vulnerable to *Four balls* rather than *Three balls*. Detailed results are included in supplementary material.



(a) *One ball* (b) *Two balls* (c) *Three balls* (d) *Four balls*

Figure 7. Rendered images of different numbers of balls.



(a) Random pixel (b) Random solid (c) Blurred image

Figure 8. Illustrations of different backgrounds.

Variation of the camera angle and distance. We conducted experiments to assess the impact of camera angle and distance variations. The attack success rate increased proportionally with the camera angle variation but dropped beyond a certain range. This finding suggests that excessive image transformation may deteriorate the transferability as the adversarial examples overfit to the source model's drastic transformation. Table 3 shows that the attack success rate increases as the range of the camera distance expands. Therefore, we expect that the more rendering parameters

[2]We included references to these 3D models in supplementary material.

(a) Dodging attack

(b) Impersonation attack

Figure 9. Illustrations of adversarial attacks with the ODI method on the face verification task.

vary, the better the transferability, within a certain range.

Different backgrounds. Finally, we examined the effect of the backgrounds. In this experiment, we employed a random solid background, random pixel values, and a blurred image. All random values were uniformly sampled between 0 and 1. For blurred images, we convolved the input image with the Gaussian kernel whose kernel size is 50 and σ is 15. According to the experimental results, the above three backgrounds performed better than the solid black background. Among all, the background with random pixel values obtained the highest attack success rate.

4.4. Adversarial Attacks on Face Recognition

The proposed technique is not limited to image classification. It can be used for adversarial attacks on other tasks as well, including face recognition [6] and object detection [29]. As an example, we applied our method to adversarial attacks on face verification models. A face verification model compares two photos to see if they are of the same person [6]. By changing an image in the pair of photos, adversaries can launch two types of attacks, which cause the same person to be classified as different people (dodging attacks) or different people to be classified as the same person (impersonation attacks) [9]. These two types of attacks with the ODI method are illustrated in Fig. 9.

We experimented with 500 face pairs in the Labeled Faces in the Wild (LFW) dataset [15] which is widely used in related works [6, 16, 31]. We used the squared ℓ_2-distance between a pair of facial features as the adversary's objective loss, $\epsilon = 8/255$ and $\epsilon = 16/255$ for dodging and impersonation attacks, respectively. We used *Pillow* as the source object and changed the range of camera angles to $[-25°, 25°]$. We used the same setting of the ImageNet dataset for all other parameters to show that a significant performance gain is attainable without expensive parameter searches. The attack success rates on three black-box models from the ArcFace ResNet-50 source model [6] are shown in Table 4. ODI-MI-TI-VT boosted the attack success rate for impersonation and dodging attacks by an average of 12.0% and 7.2%, respectively.

Impersonation attack	Target model		
Method	CurricularFace [16] RN-100 [13]	ArcFace [6] GhostNet x1.3 [12]	FaceNet [31]
DI-MI-TI	57.0	47.2	65.2
RDI-MI-TI	58.4	47.8	68.0
RDI-MI-TI-VT	61.4	51.4	71.4
ODI-MI-TI	71.8	61.8	80.6
ODI-MI-TI-VT	**73.0**	**63.0**	**84.2**
Dodging attack	Target model		
Method	CurricularFace [16] RN-100 [13]	ArcFace [6] GhostNet x1.3 [12]	FaceNet [31]
DI-MI-TI	97.2	62.4	90.8
RDI-MI-TI	96.2	68.4	90.6
RDI-MI-TI-VT	97.0	71.2	92.0
ODI-MI-TI	99.2	83.0	95.2
ODI-MI-TI-VT	**99.4**	**85.2**	**97.2**

Table 4. Transfer attack success rates (%) on the face verification models. The source model is ArcFace ResNet-50 [6].

4.5. Discussion

Limitations. Our technique is suitable for sufficiently large images that are able to preserve the original image contents with 3D models. It is hard to apply the ODI method to small images, such as the 32×32-sized images in the CIFAR-10 dataset [18], unless we expand the input image.

Potential societal impact. If the proposed ODI method is maliciously utilized, it may cause social confusion by deceiving AI-based commercial services in the world. However, our study raises awareness of this potential threat to service providers and researchers, and can contribute to developing more robust deep learning models.

Future work. Although this work focuses on rigid objects whose shapes do not change, more diverse augmentation is achievable by altering the objects' shapes. More work has to be done with this concept to increase the transferability even further. Additionally, this 3D model-based data augmentation can be used to improve the performance of deep learning models on other tasks. This technique could be especially useful when the amount of training data is limited.

5. Conclusion

In this paper, we have proposed the object-based diverse input method as a novel data augmentation technique to improve the transferability of targeted adversarial examples. Our experimental results on image classification and face verification demonstrate that the proposed method boosts the attack success rate on various black-box target models.

References

[1] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. 2020. 2

[2] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *International conference on machine learning*, pages 284–293. PMLR, 2018. 3, 4

[3] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *International Conference on Learning Representations*, 2018. 2

[4] 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*, pages 15–26, 2017. 2

[5] 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*, 2019. 2

[6] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019. 1, 8

[7] 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, 6

[8] 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*, pages 4312–4321, 2019. 2, 3, 6

[9] Yinpeng Dong, Hang Su, Baoyuan Wu, Zhifeng Li, Wei Liu, Tong Zhang, and Jun Zhu. Efficient decision-based black-box adversarial attacks on face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7714–7722, 2019. 8

[10] Andre Esteva, Alexandre Robicquet, Bharath Ramsundar, Volodymyr Kuleshov, Mark DePristo, Katherine Chou, Claire Cui, Greg Corrado, Sebastian Thrun, and Jeff Dean. A guide to deep learning in healthcare. *Nature medicine*, 25(1):24–29, 2019. 1

[11] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 2

[12] Kai Han, Yunhe Wang, Qi Tian, Jianyuan Guo, Chunjing Xu, and Chang Xu. Ghostnet: More features from cheap operations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1580–1589, 2020. 8

[13] 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. 1, 5, 8

[14] 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. 5

[15] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007. 8

[16] Yuge Huang, Yuhan Wang, Ying Tai, Xiaoming Liu, Pengcheng Shen, Shaoxin Li, Jilin Li, and Feiyue Huang. Curricularface: adaptive curriculum learning loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5901–5910, 2020. 8

[17] Hiroharu Kato, Deniz Beker, Mihai Morariu, Takahiro Ando, Toru Matsuoka, Wadim Kehl, and Adrien Gaidon. Differentiable rendering: A survey. *arXiv preprint arXiv:2006.12057*, 2020. 3

[18] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 8

[19] Alexey Kurakin, Ian Goodfellow, Samy Bengio, et al. Adversarial examples in the physical world, 2016. 2

[20] Alexey Kurakin, Ian J. Goodfellow, Samy Bengio, Yinpeng Dong, Fangzhou Liao, Ming Liang, Tianyu Pang, Jun Zhu, Xiaolin Hu, Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, Alan L. Yuille, Sangxia Huang, Yao Zhao, Yuzhe Zhao, Zhonglin Han, Junjiajia Long, Yerkebulan Berdibekov, Takuya Akiba, Seiya Tokui, and Motoki Abe. Adversarial attacks and defences competition. *CoRR*, abs/1804.00097, 2018. 5, 7

[21] Maosen Li, Cheng Deng, Tengjiao Li, Junchi Yan, Xinbo Gao, and Heng Huang. Towards transferable targeted attack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 641–649, 2020. 1, 2, 3, 4

[22] Jiadong Lin, Chuanbiao Song, Kun He, Liwei Wang, and John E Hopcroft. Nesterov accelerated gradient and scale invariance for adversarial attacks. In *International Conference on Learning Representations*, 2019. 2, 3, 6

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

[24] 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*, pages 940–949, 2020. 2

[25] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. 7

[26] 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. 1, 2

[27] 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*, pages 506–519, 2017. 2

[28] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv preprint arXiv:2007.08501*, 2020. 6

[29] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 8

[30] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 5

[31] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015. 8

[32] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5

[33] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*, 2017. 5

[34] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 5

[35] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014. 1

[36] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. In *International Conference on Learning Representations*, 2018. 7

[37] 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*, pages 1924–1933, 2021. 1, 3, 6

[38] Weibin Wu, Yuxin Su, Xixian Chen, Shenglin Zhao, Irwin King, Michael R 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*, pages 1161–1170, 2020. 2

[39] Weibin Wu, Yuxin Su, Michael R Lyu, and Irwin King. Improving the transferability of adversarial samples with adversarial transformations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9024–9033, 2021. 3, 6

[40] Chaowei Xiao, Dawei Yang, Bo Li, Jia Deng, and Mingyan Liu. Meshadv: Adversarial meshes for visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6898–6907, 2019. 3, 4

[41] 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. 2, 3, 6

[42] Xiaohui Zeng, Chenxi Liu, Yu-Siang Wang, Weichao Qiu, Lingxi Xie, Yu-Wing Tai, Chi-Keung Tang, and Alan L Yuille. Adversarial attacks beyond the image space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4302–4311, 2019. 3, 4

[43] Zhengyu Zhao, Zhuoran Liu, and Martha Larson. On success and simplicity: A second look at transferable targeted attacks. In *Advances in Neural Information Processing Systems*, 2021. 1, 2, 3, 4, 5, 6

[44] Junhua Zou, Zhisong Pan, Junyang Qiu, Xin Liu, Ting Rui, and Wei Li. Improving the transferability of adversarial examples with resized-diverse-inputs, diversity-ensemble and region fitting. In *European Conference on Computer Vision*, pages 563–579. Springer, 2020. 2, 3, 6