Exploring Transferable and Robust Adversarial Perturbation Generation from the Perspective of Network Hierarchy

Ruikui Wang[1], **Yuanfang Guo**[1], **Ruijie Yang**[1], **Yunhong Wang**[1]
[1] School of Computer Science and Engineering, Beihang University
Beijing, China
{rkwang,rjyang,andyguo,yhwang}@buaa.edu.cn

Abstract

The transferability and robustness of adversarial examples are two practical yet important properties for black-box adversarial attacks. In this paper, we explore effective mechanisms to boost both of them from the perspective of network hierarchy, where a typical network can be hierarchically divided into output stage, intermediate stage and input stage. Since over-specialization of source model, we can hardly improve the transferability and robustness of the adversarial perturbations in the output stage. Therefore, we focus on the intermediate and input stages in this paper and propose a transferable and robust adversarial perturbation generation (TRAP) method. Specifically, we propose the dynamically guided mechanism to continuously calculate accurate directional guidances for perturbation generation in the intermediate stage. In the input stage, instead of the single-form transformation augmentations adopted in the existing methods, we leverage multi-form affine transformation augmentations to further enrich the input diversity and boost the robustness and transferability of the adversarial perturbations. Extensive experiments demonstrate that our TRAP achieves impressive transferability and high robustness against certain interferences.

1 Introduction

Deep Neural Networks (DNNs) have achieved remarkable success in various computer visual tasks, including image classification [17, 34, 11], object tracking [12, 16, 20], detection [29, 26], semantic segmentation [24, 3, 5], etc. Unfortunately, these DNNs are easily to be interfered by some images, which contain certain artificial yet inconspicuous perturbations, and yield incorrect outputs. These deliberately crafted perturbations are named adversarial perturbations [38], which has attracted significant attentions of researchers in the past few years [38, 10, 19, 27, 4, 6, 7, 43]. In general, the adversarial perturbation generation (i.e., adversarial attack) methods can be classified into two categories: white-box attacks and black-box attacks. The white-box attacks usually assume that the attackers possess access to both the architecture and gradient information of the target model. On the contrary, the black-box attacks assume that the attackers possess approximately zero knowledge besides of the final predictions. In the black-box scenarios, the adversarial attack methods usually require the generated perturbations to possess high transferability [23], which is vital to the attack success rate. Recently, many literatures have been published to focusing on investigating this intriguing property [27, 6, 49, 13, 47, 14]. Besides, the property of an adversarial example, which assesses the performance drops against various interferences, is named robustness, which is also a practical concern in real world systems [2, 9, 40, 30, 35, 1, 8].

In this paper, we explore to generate adversarial perturbations with high transferability and robustness from the perspective of network hierarchy. Intuitively, a deep network architecture can be hierarchi-
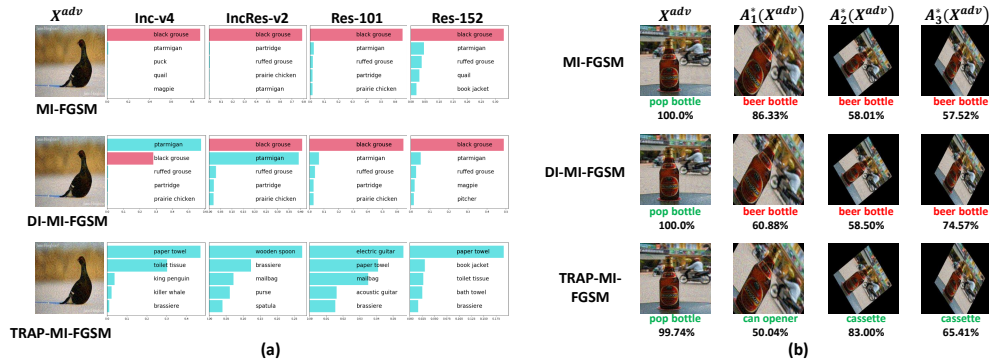
Figure 1: Demonstration of transferability and robustness of adversarial examples. All the adversarial examples are crafted on Inception-v3 with $\epsilon = 16$. (a) The comparison of transferability of adversarial examples. The ground-truth label is highlighted in pink. (b) The comparison of robustness of adversarial examples. X^{adv} denotes the original adversarial examples and $A_i^*(\cdot)$ denotes different multi-form transformations. Note that the descriptions under each image indicate its predicted category by Inception-v3 and corresponding confidence. We highlight the category tags in green if the attack is successful. If the attack is failed, the category tag is highlighted in red.

cally divided into three major stages, i.e., output stage, intermediate stage and input stage. Most of the existing adversarial attack methods manipulate in one of these stages, each of which are relatively orthogonal to the others and can thus play a complementary role to achieve higher transferability and robustness.

Unfortunately, to our best knowledge, there is no existing approaches which can manipulate in the output stage and boost the transferability. We believe that this phenomenon is induced by the over-specialization of the DNN model in the output stage, which is a classic yet typical generalization problem in machine learning. Therefore, we will not devote our efforts in this stage also, in this paper. For the intermediate stage, some literatures [33, 31, 49, 15, 13, 21, 45, 14, 25] have explored to perturb the intermediate features to improve the transferability of adversarial examples. [13] develops a new paradigm to specifically optimize the transferability by utilizing the intermediate representation of a given adversarial example as directional guidance, which provides a reasonable proxy for generating the transferable perturbations. [13] inspires [21] to further explore linear combinations of auxiliary results, which are produced in the baseline phase, as the directional guidance for the latter phase. However, since these directional guidances are fixed once the baseline phase is completed, the positive effects of these guidances tend to decline as the subsequent optimization steps carry out. Meanwhile, the transferability, as well as the robustness, of the adversarial examples benefits from creating diverse inputs at the input stage of network [2, 22, 47, 50, 48]. [2] generates robust adversarial examples by introducing Expectation Over Transformation (EOT) to the inputs. Unfortunately, [2] only applies single-form transformations, i.e. only one type of basic transformations at a time, to the input image, where the complete (multi-form) affine transformation cannot be constructed to improve the transferability and robustness of the adversarial examples. Note that, the expectation operation tends to bring larger computational overheads.

To tackle these issues, we propose a transferable and robust adversarial perturbation generation (TRAP) method from the perspective of network hierarchy. Specifically, dynamically guided mechanism, which adaptively updates the directional guidance as the optimization progresses, is proposed to boost the transferability in the intermediate stage. Figure 1(a) gives an example of the transferability property of our method. In the input stage, we introduce a multi-form affine-transformation, which contains multiple types of basic transformations, to the adversarial examples to perform combinatorial augmentations with little computational overheads. Figure 1(b) reveals that our primary intention of utilizing the multi-form transformation augmentation is to boost the robustness of the perturbations to the practical variations.

Our contributions are summarized as below:

- We propose a transferable and robust adversarial perturbation generation (TRAP) method from the perspective of network hierarchy to boost the transferability and robustness of the adversarial examples simultaneously.
- We propose a dynamically guided mechanism in the intermediate stage of network to adaptively revise the directional guidance, as the perturbation generation process performs.
- We propose an affine-invariant perturbation enhancement mechanism, which improves both the transferability and robustness of the adversarial perturbations, by augmenting the input images with multi-form affine transformations, in the input stage.

2 Related Work

In this section, we give a brief review to the related black-box attack methods. By regarding the adversarial perturbation generation process as an optimization problem, gradient-based methods usually leverage the existing optimization algorithms to boost the transferability of adversarial examples [10, 19, 18, 39, 6, 22, 42].

For the intermediate stage of network, perturbing the intermediate feature space is proposed to improve the transferability of the black-box methods. Transferable Adversarial Perturbations (TAP) [49] maximizes the distances between the benign images and their adversarial versions at all the hidden layers. To search for perturbations with better transferability, Intermediate Level Attack (ILA) [13] maximizes the scalar projection of the adversarial example onto a guided direction on a specific hidden layer. Motivated by ILA [13], [21] takes the advantage of auxiliary examples produced by a baseline attack and yields adversarial examples with better transferability.

In the input stage of network, input augmentation can be exploited to facilitate both the robustness and transferability. EOT [2] adopts this principle and generates robust adversarial examples via single-form affine transformation augmentations. Diverse Input Method (DIM) [47] further demonstrates the effectiveness of input augmentation by applying random resizing and padding to the inputs with a fixed probability. Scale-Invariant Method (SIM) [22] enriches the gradient information over an ensemble of multi-scale copies of input image and generates more transferable adversarial examples.

3 Proposed Work

Given a clean image X, its ground-truth label y^{true} and a substitute network with parameters θ, we aim to generate an adversarial perturbation for X with high transferability and robustness. Considering the relative orthogonality of different stages, naturally, we can design different mechanisms for each stage and combine them to construct a novel adversarial perturbation generation method, named transferable and robust adversarial perturbation generation (TRAP). The complete procedures of our TRAP are summarized in Algorithm 1, which contains two mechanisms, i.e., *dynamically guided mechanism* and *affine-invariant perturbation enhancement mechanism*. Note that the baseline attack model in our method can be any transfer-based black-box attack methods with any gradient based iterative optimizer. For convenience, we simply employ MI-FGSM [6] as our baseline model and Momentum-SGD [28] as the optimizer in this paper.

3.1 Dynamically Guided Mechanism (DGM)

For the intermediate stage of network, [21] has empirically demonstrated that a larger perturbation on the intermediate feature leads to a higher transferability. Then, a straightforward strategy is to explicitly maximize the distances between the benign images and their corresponding adversarial examples in feature space [49, 45, 15]. However, the transferability of such straightforwardly generated examples usually becomes unsatisfactory, which is induced by overfitting the source model when the number of attack iteration increases. On the contrary, [13, 21] leverage two phases, i.e., *baseline phase* and *enhancement phase*, to implicitly enlarge the intermediate feature gap between the benign and perturbed images, with respect to a directional guidance obtained in the baseline phase. Unfortunately, [13, 21] only exploit fixed directional guidance, which is generated by the baseline phase and can only provide sufficient guidelines to the initial steps of the enhancement phase. It cannot provide accurate reference information for subsequent optimization procedures. Therefore, we propose a dynamically guided mechanism to alleviate this problem.

Algorithm 1 Transferable and Robust Adversarial Perturbation Generation (TRAP)

Input: A benign example X with ground-truth label y^{true}; the source model parameter θ;
Input: Perturbation budget ϵ; maximum iterations T; iterations of baseline attack phase t_1; momentum factor μ; the transformation probability p;
Output: Final adversarial example X_T^{adv}

1: $\alpha = \epsilon/t_1$; $X_0^{adv} = X$
2: $g_m = 0$; $h_0^* = 0$
3: **for** $t = 0$ to $T - 1$ **do**
4:     **if** $t == t_1$ **then**
5:         $X_t^{adv} = X$; $g_m = 0$
6:     **if** $t < t_1$ **then**
7:         $g = \nabla_X L_1(A_1 \circ A_2 \circ A_3 \circ A_4(X_t^{adv}; p), y^{true}; \theta)$
8:     **else**
9:         $g = \nabla_X L_2(A_1 \circ A_2 \circ A_3 \circ A_4(X_t^{adv}; p), h_t^*; \theta)$
10:     Update $g_{t+1} = \mu \cdot g_m + \frac{g}{\|g\|_1}$; $g_m = g_{t+1}$
11:     **if** $t >= t_1$ **then**
12:         $\alpha = \epsilon/(T - t_1)$
13:     Update $X_{t+1}^{adv} = Clip_X^\varepsilon \{X_t^{adv} + \alpha \cdot sign(g_{t+1})\}$
14:     **if** $t >= t_1$ **then**
15:         Update h_{t+1}^* by Eq.3
16:     **else**
17:         $h_{t+1}^* = flatten(\mathscr{F}^l(X_{t+1}^{adv}; \theta))$
18: **return** X_T^{adv}

Specifically, we firstly perform t_1 steps of the baseline attack phase and adopt cross-entropy loss to train initial the attack model as:

$$L_1(X^{adv}, y^{true}; \theta) = -\mathbb{1}_{y^{true}} log(softmax(c(X^{adv}; \theta))), \tag{1}$$

where $\mathbb{1}_{y^{true}}$ represents the one-hot formed ground-truth and $c(X; \theta)$ denotes the logits output and θ stands for the network parameters. Then, an initial adversarial example $X_{t_1}^{adv}$ can be obtained. In our enhancement phase, we progressively update the directional guidance as the enhancement process continues. In practice, we initialize the directional guidance $h_{t_1}^*$ with the hidden output $h_{t_1}^{adv}$, which can be computed by

$$h_{t_1}^{adv} = flatten(\mathscr{F}^l(X_{t_1}^{adv}; \theta)), \tag{2}$$

where $\mathscr{F}^l(\cdot; \theta)$ indicates the output function of layer l of source model. Then, it can be updated in a progressive manner via

$$h_t^* = (1 - \beta)h_{t-1}^{adv} + \beta h_{t-1}^*, t \geq t_1, \tag{3}$$

where t denotes the optimization step in the enhancement phase, h_t^{adv} represents the hidden output of X_t^{adv}, and β is employed to balance the tradeoff between the historical and current directional guidances. Then, the gradient optimization direction can be sought for, with the help of the above dynamically updated directional guidance. Similar to [13], we not only expect the optimized direction h_t^{adv} to be consistent with the dynamical directional guidance h_t^*, but also anticipate a larger amplitude of the perturbation in that direction. This optimization can be achieved via:

$$L_2(X_t^{adv}, h_t^*; \theta) = \frac{< h_t^* - h^x, h_t^{adv} - h^x >}{\|h_t^* - h^x\|_2 \|h_t^{adv} - h^x\|_2} + \gamma \frac{\|h_t^{adv} - h^x\|_2}{\|h_t^* - h^x\|_2}, t \geq t_1, \tag{4}$$

where h^x and h_t^{adv} stand for the hidden outputs of the original image X and X_t^{adv}, respectively, and γ denotes the tradeoff parameter.

3.2 Affine-Invariant Perturbation Enhancement Mechanism (AIM)

For input stage of network, we introduce a new augmentation paradigm, i.e., imposing multiple basic instantiations of affine transformations on input concurrently to enrich the diversity of input patterns, when generating the adversarial examples. Specifically, we leverage four types of basic

transformations, including translation, rotation, scaling and shearing. Note that these differentiable basic transformations can be formulated in an unified mathematical expression and the calculations can benefit from GPU accelerations. Besides, the invariance against these transformations is of great significance to facilitate the robustness of adversarial examples [2].

Let $A_1(\cdot)$, $A_2(\cdot)$, $A_3(\cdot)$, $A_4(\cdot)$ denote the translation, rotation, scaling and shearing operations respectively. In this paper, these operations are formulated in a manner of coordinate transformation. Specifically, the formulation of translation, $A_1(\cdot)$, can be defined as

$$\left[\begin{array}{c} x' \\ y' \\ 1 \end{array} \right] = \left[\begin{array}{ccc} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{array} \right] \left[\begin{array}{c} x \\ y \\ 1 \end{array} \right], \tag{5}$$

where (x, y) denotes the coordinates of the original image pixel, (x', y') denotes the coordinates of the transformed image pixel, and t_x and t_y represent the offsets of translation in the horizontal and vertical directions, respectively. Similarly, the formulation of rotation, $A_2(\cdot)$, can be defined as

$$\left[\begin{array}{c} x' \\ y' \\ 1 \end{array} \right] = \left[\begin{array}{ccc} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{array} \right] \left[\begin{array}{c} x \\ y \\ 1 \end{array} \right], \tag{6}$$

where θ represents the degree of rotation. The formulation of scaling, $A_3(\cdot)$, can be defined as

$$\left[\begin{array}{c} x' \\ y' \\ 1 \end{array} \right] = \left[\begin{array}{ccc} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{array} \right] \left[\begin{array}{c} x \\ y \\ 1 \end{array} \right], \tag{7}$$

where s_x and s_y denote the scaling factors of the horizontal and vertical directions, respectively. The formulation of shearing, $A_4(\cdot)$, can be defined as

$$\left[\begin{array}{c} x' \\ y' \\ 1 \end{array} \right] = \left[\begin{array}{ccc} 1 & d_x & 0 \\ d_y & 1 & 0 \\ 0 & 0 & 1 \end{array} \right] \left[\begin{array}{c} x \\ y \\ 1 \end{array} \right], \tag{8}$$

where d_x and d_y denote the two shearing factors.

With the above defined basic transformations in (5) to (8), a multi-form affine transformation can be achieved simply by multiplying these transformation matrices. In practice, we impose multi-form transformations on adversarial example generated in each step. The updating rules of our proposed AIM can be formulated in such a iterative manner:

$$X_{t+1}^{adv} = Clip_X^{\epsilon}(X_t^{adv} + \alpha \cdot sign(\nabla_X L(A_1 \circ A_2 \circ A_3 \circ A_4(X_t^{adv}; p), y^{true}; \theta))), \tag{9}$$

where $Clip_X^{\epsilon}(\cdot)$ denotes the clip operation which ensures the generated perturbed image is within the $\epsilon-$ball of the benign image X. Note that $\nabla_X L(X, y^{true}; \theta)$ represents the gradient of the final loss with respect to X. $A_1 \circ A_2 \circ A_3 \circ A_4$ stands for our multi-form affine transformation operation, which can degenerate to the single-form one by fixing certain transformation factors or offsets. Similar to [47], p controls the execution probability of applying affine-transformation in each iteration. t and α denote the number of iterations and step size, respectively.

4 Experimental Results

In this section, we evaluate our proposed TRAP in various experiments. Please note that we employ the prefix 'DG-' to represent the utilization of our dynamically guided mechanism and 'AI-' to represent the utilization of our affine-invariant perturbation enhancement mechanism, in the results.

4.1 Experimental Settings

Dataset. By following the experimental protocols in [22, 41], we randomly select 1000 images, which belong to 1000 classes, from the ILSVRC2012 validation set [32]. For performance measurement, we adopt the commonly used attack success rate (ASR).

Networks. We adopt five DNN models, i.e., Inception-v3(Inc-v3) [37], Inception-v4(Inc-v4) [36], Inception-Resnet-v2(IncRes-v2) [36], ResNet-101(Res-101) [11], ResNet-152(Res-152) [11]. All the models are available on Github[1].
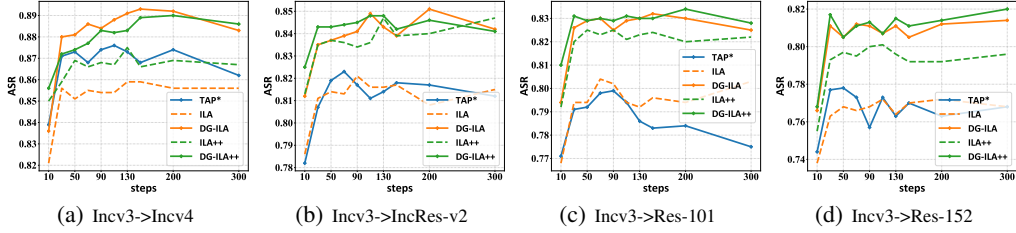
---
[1]https://github.com/Cadene/pretrained-models.pytorch

(a) Incv3->Incv4 (b) Incv3->IncRes-v2 (c) Incv3->Res-101 (d) Incv3->Res-152

Figure 2: Results of DGM when varying the number of iterations T. The caption 'A'->'B' in each sub-figure indicates that 'A' is the source model and 'B' is the target model.

Implementation details. In the experiments, MI-FGSM is employed as our baseline model. For fair comparisons, all of the methods in the experiments employ Momentum-SGD as the optimizer and the momentum factor μ is set to 1.0. The maximum perturbation for each pixel is set to be $\epsilon = 16$. The number of iterations T is set to 10. And step size is determined by $\alpha = \epsilon/T$. For DI-MI-FGSM [47] and AI-MI-FGSM, the transformation probability p is set to 0.9. For ILA [13] and DG-ILA, we set γ to 0.8. For DG-ILA, we set β to 0.8 and the number of iterations for the baseline phase t_1 to 4. All the input images are resized to $[299, 299, 3]$. As for affine-transformation, we determine t_x, t_y by sampling from the uniform distribution $[-0.1, 0.1]$ for each step. Similar, θ is sampled from $[-90, 90]$, s_x, s_y are sampled from $[0.5, 1.5]$ and d_x, d_y are sampled from $[-30, 30]$.

4.2 Evaluation of DGM

In this subsection, we focus on evaluating our proposed DGM. Following the protocols in [13], the same intermediate layer are selected for all the methods. Specifically, the selected intermediate layer for Inc-v3, Inc-v4, IncRes-v2, Res-101 and Res-152 are 'Mixed6c', 'feature-9', 'mixed6a', 'layer3' and 'layer2', respectively. According to our experiments, the performance of constraining all the layers in original TAP [49] is actually lower than that of constraining certain layer alone, which is implemented by us and denoted as TAP*. Besides, the results of ILA [13] and ILA++ [21] are reproduced with their released source codes.

Ablation Study. The results of ablation study are given in Table 1. As can be observed, both DG-ILA and DG-ILA++ obviously outperforms their original baseline methods. When the source model is IncRes-v2, the performance gain can reach as much as 5%. Besides, DG-ILA performs better than another related work TAP* in most cases and DG-ILA++ can further improve the performance. Apparently, our DGM can successfully boost the transferability of black-box attack methods.

Table 1: The evaluation of our DGM. The source models we used are listed in the first column and the target models are listed in the first line.

Model	Attack	Inc-v3	Inc-v4	IncRes-v2	Res-101	Res-152
Inc-v3	TAP[49]	99.8%	81.1%	75.9%	77.6%	73.6%
	TAP*	99.7%	83.9%	78.2%	77.1%	74.4%
	ILA[13]	99.6%	82.1%	78.6%	76.8%	73.8%
	DG-ILA(ours)	99.7%	83.6%	81.2%	79.4%	76.6%
	ILA++[21]	99.7%	85.0%	81.3%	79.2%	75.5%
	DG-ILA++(ours)	99.5%	85.6%	82.5%	81.0%	76.8%
Inc-v4	TAP[49]	52.8%	99.9%	45.3%	48.2%	43.2%
	TAP*	79.2%	98.7%	70.3%	76.2%	73.4%
	ILA[13]	77.9%	98.2%	71.3%	74.7%	70.4%
	DG-ILA(ours)	78.7%	98.0%	72.8%	77.7%	73.1%
	ILA++[21]	78.9%	99.0%	73.3%	77.9%	73.2%
	DG-ILA++(ours)	80.2%	98.8%	74.8%	78.7%	74.9%
IncRes-v2	TAP[49]	62.8%	59.6%	96.0%	63.6%	56.8%
	TAP*	79.1%	74.9%	97.1%	73.0%	69.0%
	ILA[13]	76.5%	74.1%	97.0%	73.1%	68.5%
	DG-ILA(ours)	82.1%	78.4%	97.5%	76.9%	72.3%
	ILA++[21]	79.5%	76.1%	98.1%	75.6%	69.9%
	DG-ILA++(ours)	82.7%	78.1%	98.1%	78.0%	72.7%
Res-101	TAP[49]	63.0%	56.6%	50.4%	100.0%	92.8%
	TAP*	77.6%	75.2%	65.2%	100.0%	98.8%
	ILA[13]	72.3%	69.5%	61.2%	100.0%	97.7%
	DG-ILA(ours)	74.1%	71.7%	62.9%	100.0%	97.5%
	ILA++[21]	75.6%	72.0%	65.1%	100.0%	98.3%
	DG-ILA++(ours)	77.7%	74.5%	67.1%	100.0%	98.5%
Res-152	TAP[49]	63.9%	58.2%	53.6%	94.9%	100.0%
	TAP*	72.1%	70.9%	63.7%	94.8%	99.2%
	ILA[13]	73.7%	73.8%	67.9%	96.2%	99.7%
	DG-ILA(ours)	76.6%	77.4%	68.5%	96.2%	98.7%
	ILA++[21]	74.6%	75.5%	68.0%	96.7%	99.8%
	DG-ILA++(ours)	76.5%	78.2%	68.9%	96.5%	99.2%

Effects of the Number of Iterations. Figure 2 presents the steps-vs-ASR results. We can observe that DG-ILA has a sustainable advantage over ILA as the increase of iteration number. Similar tendency can also be seen from ILA++ and DG-ILA++. A possible explanation is that the current search direction, which is usually better than the guidance from baseline attack phase, will make contribution for next search and such progressive manner has more potential to find transferable directions. As for TAP*, it may fall into overfitting more easily because of fixed optimization objectives.

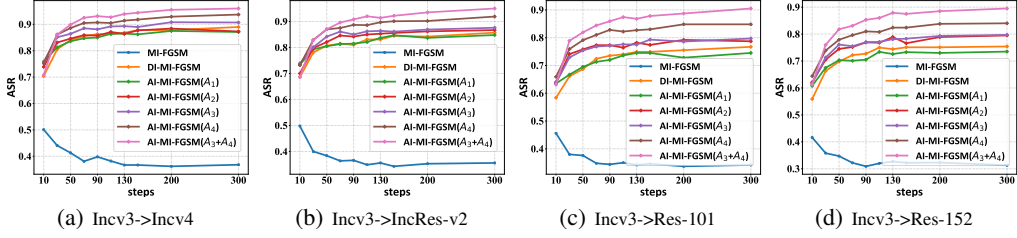| (a) Incv3->Incv4 | (b) Incv3->IncRes-v2 | (c) Incv3->Res-101 | (d) Incv3->Res-152 |

Figure 4: Evaluation results of AIM and superiority of multi-form transformation when varying the iteration number T. The sub-script 'A'->'B' below each sub-plot represents that A is the source model and B is the target model.

Effects of Different Layer Selections. Since ILA is a layer-centric attack method, the to-be-manipulated layer is specifically selected on the source model to obtain the best performance on the target model. Then, it is necessary to examine the performance of our DGM across various layers. Taking Inception-v3 as the source model, we select the layers from 'Conv2d_1a_3x3' to 'last_linear', except for the pooling and dropout layers, as the intermediate layers for evaluation. The results are shown in Figure 3(a). As can be observed, DG-ILA performs better than ILA in most of the selected layers and can give the best performance in black-box scenarios. Note that

Figure 3: (a) The attack success rate comparison across various layers. (b) The relative differences comparison of intermediate features.

DG-ILA does not function very well at the bottom and top layers, which may be induced by the initial directional guidance is less transferable and lacks basic guidance ability. Fortunately, these layers will not be selected in practice because of their poor ASRs compared to the mid-level layers. To further verify the effectiveness of our DGM, we also calculate the relative feature difference, which is computed by $\|\mathscr{F}^l(X^{adv}) - \mathscr{F}^l(X)\|_2/\|\mathscr{F}^l(X)\|_2$, between the benign and the corresponding adversarial examples across various layers on Inception-v3. As can be observed from Figure 3(b), DG-ILA gives larger mid-layer disturbance (in feature maps) than ILA, which is consistent with our expectation and the conclusion drawn in [49] and [21].
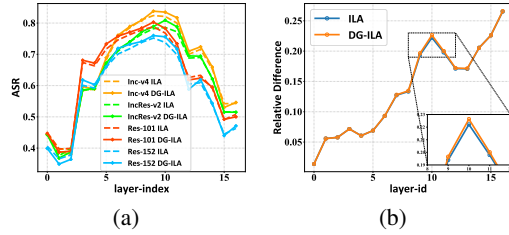
4.3 Evaluation of AIM

Ablation Study. We compare three attack methods, MI-FGSM [6], DI-MI-FGSM (momentum version of DIM) [47] and our AI-MI-FGSM, and present quantitative results in Table 2. Note that A_i denotes the employed single-form transformation A_i defined in 5 to 8. From Table 2, a first glance shows that DI-MI-FGSM outperforms MI-FGSM by a large margin and AI-MI-FGSM is comparable to DI-MI-FGSM and even better than it in most black-box settings. This table verifies the usefulness of our each single-form operation when boosting the transferability of adversarial examples.

Multi-form Operation Analysis. To further validate the effectiveness of the multi-form affine transformation augmentation, Inception-v3 is selected and the number of iterations are varied. The results are presented in Figure 4. As can be observed, if the multi-form transfor-

Table 2: The evaluation of AIM and usefulness of single-form transformation adopted by us.

Model	Attack	Inc-v3	Inc-v4	IncRes-v2	Res-101	Res-152
Inc-v3	MI-FGSM[6]	**100.0%**	51.3%	49.2%	46.8%	42.4%
	DI-MI-FGSM[47]	98.2%	72.3%	69.5%	59.2%	58.1%
	AI-MI-FGSM(A_1)	99.3%	75.5%	72.5%	61.9%	60.6%
	AI-MI-FGSM(A_2)	96.3%	73.8%	72.0%	63.3%	62.4%
	AI-MI-FGSM(A_3)	97.7%	76.6%	**74.0%**	**65.2%**	**64.5%**
	AI-MI-FGSM(A_4)	98.3%	**76.8%**	73.3%	63.7%	64.3%
Inc-v4	MI-FGSM[6]	53.5%	**100.0%**	47.3%	45.9%	41.8%
	DI-MI-FGSM[47]	72.4%	96.0%	69.1%	59.2%	59.1%
	AI-MI-FGSM(A_1)	**76.9%**	97.7%	**72.2%**	63.2%	61.4%
	AI-MI-FGSM(A_2)	73.2%	93.0%	69.9%	63.7%	60.8%
	AI-MI-FGSM(A_3)	76.1%	96.4%	70.9%	**66.0%**	**64.8%**
	AI-MI-FGSM(A_4)	76.7%	94.0%	70.9%	64.3%	62.5%
IncRes-v2	MI-FGSM[6]	58.6%	51.8%	**98.2%**	46.6%	45.8%
	DI-MI-FGSM[47]	69.5%	65.4%	87.6%	55.7%	55.9%
	AI-MI-FGSM(A_1)	**73.9%**	**72.4%**	91.2%	61.3%	62.0%
	AI-MI-FGSM(A_2)	72.6%	71.5%	88.0%	63.4%	63.6%
	AI-MI-FGSM(A_3)	**73.9%**	69.7%	88.8%	**65.0%**	**64.6%**
	AI-MI-FGSM(A_4)	70.1%	70.0%	87.4%	61.3%	62.0%
Res-101	MI-FGSM[6]	53.1%	47.3%	39.3%	**100.0%**	90.5%
	DI-MI-FGSM[47]	77.7%	73.5%	67.8%	**100.0%**	94.7%
	AI-MI-FGSM(A_1)	83.9%	**82.4%**	**77.8%**	**100.0%**	**97.9%**
	AI-MI-FGSM(A_2)	81.1%	77.6%	70.5%	99.3%	93.3%
	AI-MI-FGSM(A_3)	80.4%	78.0%	74.1%	99.5%	94.6%
	AI-MI-FGSM(A_4)	**82.2%**	79.6%	74.8%	99.7%	94.1%
Res-152	MI-FGSM[6]	53.6%	49.4%	41.9%	92.1%	**100.0%**
	DI-MI-FGSM[47]	78.7%	76.6%	71.4%	96.1%	**100.0%**
	AI-MI-FGSM(A_1)	**84.4%**	**82.6%**	**79.1%**	96.7%	**100.0%**
	AI-MI-FGSM(A_2)	81.6%	78.1%	74.2%	94.9%	99.3%
	AI-MI-FGSM(A_3)	81.3%	79.6%	73.2%	**96.9%**	99.7%
	AI-MI-FGSM(A_4)	83.3%	80.5%	76.9%	95.2%	99.8%

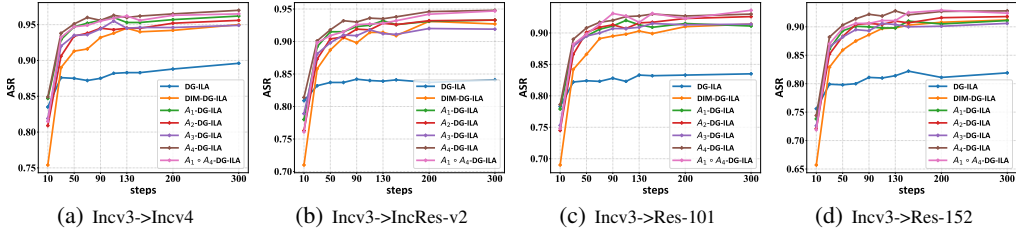| (a) Incv3->Incv4 | (b) Incv3->IncRes-v2 | (c) Incv3->Res-101 | (d) Incv3->Res-152 |

Figure 6: Combination results of DG-ILA and various input transformation based methods. The caption 'A'->'B' in each sub-figure represents that 'A' is the source model and 'B' is the target model.

mation is utilized, e.g., 'scaling+shearing', the attack success rates are superior for most of the iterations. Note that the performance of the multi-form transformation at the beginning is lower than the single-form transformations, which may be induced by the augmented input space with more diversity. Besides, MI-FGSM [6] tends to fall into the overfitting dilemma when the number of iterations become large, while DI-MI-FGSM [47] can slightly alleviate this problem. In general, it is undeniable that AI-MI-FGSM with multi-form transformation surpasses DI-MI-FGSM [47] and further boost the transferability.

Robustness Analysis. To evaluate the robustness of the generated adversarial examples from our method, we adopt Gaussian noise and Gaussian blurring to corrupt the generated adversarial examples to observe their attacking performances. Here, we uses the destruction rate, which is proposed in [19, 44], to quantify the robustness. Without loss of generality, we select Inception-v3 and Res-101 as the source and target models respectively. 1000 images, which is the same as previous experiments, are still utilized. The destruction results under Gaussian blurring and Gaussian noise are shown in Figure 5(a) and Figure 5(b) respectively. As can be observed, the adversarial examples synthesized by AI-MI-FGSM are more robust than the baseline methods. Note that our multi-form transformation, e.g. 'scaling+shearing', obviously boosts the performance.
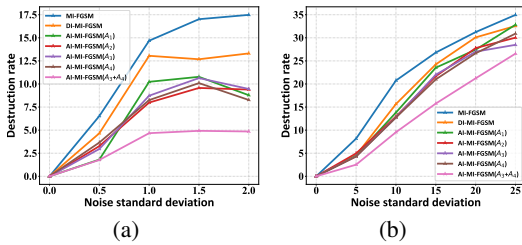

Figure 5: Comparison of destruction rate for various methods under corruption. The lower the curve, the more robust against Gaussian corruption.

4.4 Evaluation of TRAP

Effects of the Combination of DGM and AIM. To verify the effectiveness of combining the two proposed mechanisms, i.e., DGM and AIM, we select different input transformations and execute Algorithm 1 to generate the final adversarial examples. The results are shown in Figure 6. As can be observed, the obvious performance improvement indicates that our two mechanisms can collaborate complementarily, even with single basic transformations. If we exploit the multi-form affine transformation in the combination, the results are comparative with the best performance of single-form and even better, especially on Res-101.

Comparison with State-of-the-Arts. Here, the proposed TRAP is compared to the existing SOTAs on five models, which are employed in the previous experiments, as well as three defense models, i.e., adv-ResNet152 Baseline (adv-Res152B), adv-ResNet152 Denoise (adv-Res152D), adv-ResNeXt101 DenoiseAll (adv-ResNeXtDA) [46]. Note that we have subtracted the ratio of wrongly predicted benign images in the reported ASR results. Because these models are very robust and in order to reach full convergence, we perform 300 steps for all comparative attack methods. As can be observed from Table 3, our TRAP significantly outperforms the existing SOTAs on the normally trained models in most of the cases. Meanwhile, our TRAP gives better performance on the defended models in most of the cases.

8

Table 3: The ASRs of our TRAP and other SOTAs on five normally trained models and three defense models.

Model	Attack	Inc-v3	Inc-v4	IncRes-v2	Res-101	Res-152	adv-Res152B	adv-Res152D	adv-ResNeXtDA
Inc-v3	MI-FGSM[6]	**100.0%**	36.8%	35.6%	34.2%	31.3%	0.8%	0.7%	0.7%
	DI-MI-FGSM[47]	**100.0%**	89.0%	85.7%	76.7%	75.4%	2.7%	2.7%	2.9%
	TI-MI-FGSM[7]	**100.0%**	40.0%	36.9%	35.9%	33.9%	0.4%	0.8%	1.1%
	TAP[49]	99.7%	86.2%	81.2%	77.5%	76.8%	1.9%	2.4%	2.1%
	ILA[13]	99.8%	85.6%	81.5%	80.3%	76.8%	2.3%	2.5%	2.4%
	ILA++[21]	99.8%	86.7%	84.7%	82.2%	79.6%	2.5%	2.8%	2.7%
	TRAP(ours)	99.7%	**96.5%**	**94.7%**	**93.6%**	**92.4%**	**3.7%**	**4.6%**	**3.1%**
Res-101	MI-FGSM[6]	44.2%	37.6%	31.3%	**100.0%**	86.3%	0.9%	1.1%	1.3%
	DI-MI-FGSM[47]	**93.5%**	93.6%	88.7%	**100.0%**	**100.0%**	2.6%	3.1%	**4.4%**
	TI-MI-FGSM[7]	52.3%	46.0%	40.2%	**100.0%**	88.8%	1.4%	1.7%	2.8%
	TAP[49]	76.3%	74.9%	62.1%	**100.0%**	98.9%	0.7%	1.6%	1.6%
	ILA[13]	75.1%	73.1%	65.3%	**100.0%**	98.7%	1.1%	1.7%	2.4%
	ILA++[21]	77.5%	75.5%	68.1%	**100.0%**	99.2%	1.3%	1.7%	2.7%
	TRAP(ours)	93.0%	**93.8%**	**89.6%**	**100.0%**	**100.0%**	**3.8%**	**3.5%**	3.4%

5   Conclusion

In this work, we investigate to improve the transferability and robustness of adversarial examples from the perspective of network hierarchy. In the intermediate stage of network, we propose a dynamically guided mechanism to iteratively revising the directional guidance during the perturbation generation process. In the input stage of network, we propose to adopt multi-form affine transformation to augment the input images to enrich the input diversity. Experimental results have demonstrated the effectiveness of our proposed DGM and AIM, as well as the transferability and robustness of our proposed TRAP.

References

[1] A. Arnab, O. Miksik, and P. H. Torr. On the robustness of semantic segmentation models to adversarial attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 888–897, 2018.

[2] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok. Synthesizing robust adversarial examples. In *International Conference on Machine Learning*, pages 284–293, 2018.

[3] M. Bucher, T.-H. Vu, M. Cord, and P. Pérez. Zero-shot semantic segmentation. In *Advances in Neural Information Processing Systems*, 2019.

[4] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy,*, pages 39–57, 2017.

[5] K. Chaitanya, E. Erdil, N. Karani, and E. Konukoglu. Contrastive learning of global and local features for medical image segmentation with limited annotations. In *Advances in Neural Information Processing Systems*, 2020.

[6] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9185–9193, 2018.

[7] Y. Dong, T. Pang, H. Su, and J. 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.

[8] L. Engstrom, B. Tran, D. Tsipras, L. Schmidt, and A. Madry. Exploring the landscape of spatial robustness. In *International Conference on Machine Learning*, pages 1802–1811, 2019.

[9] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. 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.

[10] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2014.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *Proceedings of the European Conference on Computer Vision*, pages 630–645, 2016.

[12] D. Held, S. Thrun, and S. Savarese. Learning to track at 100 fps with deep regression networks. In *European conference on computer vision*, pages 749–765. Springer, 2016.

[13] Q. Huang, I. Katsman, H. He, Z. Gu, S. Belongie, and S.-N. Lim. Enhancing adversarial example transferability with an intermediate level attack. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4733–4742, 2019.

[14] N. Inkawhich, K. J. Liang, L. Carin, and Y. Chen. Transferable perturbations of deep feature distributions. In *International Conference on Learning Representations*, 2020.

[15] N. Inkawhich, W. Wen, H. H. Li, and Y. 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.

[16] A. R. Kosiorek, A. Bewley, and I. Posner. Hierarchical attentive recurrent tracking. In *Advances in Neural Information Processing Systems*, 2017.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[18] A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial machine learning at scale. In *International Conference on Learning Representations*, 2017.

[19] A. Kurakin, I. Goodfellow, S. Bengio, et al. Adversarial examples in the physical world. In *International Conference on Learning Representations (workshop)*, 2017.

[20] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4282–4291, 2019.

[21] Q. Li, Y. Guo, and H. Chen. Yet another intermediate-level attack. In *Proceedings of European Conference on Computer Vision*, pages 241–257, 2020.

[22] J. Lin, C. Song, K. He, L. Wang, and J. E. Hopcroft. Nesterov accelerated gradient and scale invariance for adversarial attacks. In *International Conference on Learning Representations*, 2020.

[23] Y. Liu, X. Chen, C. Liu, and D. Song. Delving into transferable adversarial examples and black-box attacks. In *International Conference on Learning Representations*, 2017.

[24] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[25] Y. Lu, Y. Jia, J. Wang, B. Li, W. Chai, L. Carin, and S. 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.

[26] T. Pang, C. Du, Y. Dong, and J. Zhu. Towards robust detection of adversarial examples. In *Advances in Neural Information Processing Systems*, 2017.

[27] N. Papernot, P. McDaniel, and I. Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *CoRR*, 2016.

[28] B. T. Polyak. Some methods of speeding up the convergence of iteration methods. *Ussr computational mathematics and mathematical physics*, 4(5):1–17, 1964.

[29] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, 2015.

[30] A. Ross and F. Doshi-Velez. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[31] A. Rozsa, M. Günther, and T. E. Boult. Lots about attacking deep features. In *IEEE International Joint Conference on Biometrics (IJCB)*, pages 168–176, 2017.

[32] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[33] S. Sabour, Y. Cao, F. Faghri, and D. J. Fleet. Adversarial manipulation of deep representations. In *International Conference on Learning Representations*, 2016.

[34] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[35] D. Song, K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, F. Tramer, A. Prakash, and T. Kohno. Physical adversarial examples for object detectors. In *12th {USENIX} Workshop on Offensive Technologies ({WOOT} 18)*, 2018.

[36] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017.

[37] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.

[38] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014.

[39] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel. Ensemble adversarial training: Attacks and defenses. In *International Conference on Learning Representations*, 2018.

[40] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry. Robustness may be at odds with accuracy. 2019.

[41] X. Wang and K. He. Enhancing the transferability of adversarial attacks through variance tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[42] X. Wang, J. Lin, H. Hu, J. Wang, and K. He. Boosting adversarial transferability through enhanced momentum. *arXiv preprint arXiv:2103.10609*, 2021.

[43] X. Wang, J. Ren, S. Lin, X. Zhu, Y. Wang, and Q. Zhang. A unified approach to interpreting and boosting adversarial transferability. In *International Conference on Learning Representations*, 2020.

[44] L. Wu, Z. Zhu, C. Tai, et al. Understanding and enhancing the transferability of adversarial examples. *arXiv preprint arXiv:1802.09707*, 2018.

[45] W. Wu, Y. Su, X. Chen, S. Zhao, I. King, M. R. Lyu, and Y.-W. 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.

[46] C. Xie, Y. Wu, L. v. d. Maaten, A. L. Yuille, and K. He. Feature denoising for improving adversarial robustness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 501–509, 2019.

[47] C. Xie, Z. Zhang, Y. Zhou, S. Bai, J. Wang, Z. Ren, and A. 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.

[48] B. Yang, H. Zhang, Y. Zhang, K. Xu, and J. Wang. Adversarial example generation with adabelief optimizer and crop invariance. *arXiv preprint arXiv:2102.03726*, 2021.

[49] W. Zhou, X. Hou, Y. Chen, M. Tang, X. Huang, X. Gan, and Y. Yang. Transferable adversarial perturbations. In *Proceedings of European Conference on Computer Vision*, pages 452–467, 2018.

[50] J. Zou, Z. Pan, J. Qiu, X. Liu, T. Rui, and W. Li. Improving the transferability of adversarial examples with resized-diverse-inputs, diversity-ensemble and region fitting. In *Proceedings of European Conference on Computer Vision*, pages 563–579, 2020.