Adaptive Image Transformations for Transfer-based Adversarial Attack

Zheng Yuan[1,2], Jie Zhang[1,2,3], and Shiguang Shan[1,2]

[1] Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
[2] University of Chinese Academy of Sciences, Beijing, China
[3] Institute of Intelligent Computing Technology, Suzhou, CAS, Suzhou, China
zheng.yuan@vipl.ict.ac.cn {zhangjie,sgshan}@ict.ac.cn

Abstract. Adversarial attacks provide a good way to study the robustness of deep learning models. One category of methods in transfer-based black-box attack utilizes several image transformation operations to improve the transferability of adversarial examples, which is effective, but fails to take the specific characteristic of the input image into consideration. In this work, we propose a novel architecture, called Adaptive Image Transformation Learner (AITL), which incorporates different image transformation operations into a unified framework to further improve the transferability of adversarial examples. Unlike the fixed combinational transformations used in existing works, our elaborately designed transformation learner adaptively selects the most effective combination of image transformations specific to the input image. Extensive experiments on ImageNet demonstrate that our method significantly improves the attack success rates on both normally trained models and defense models under various settings.

Keywords: Adversarial Attack, Transfer-based Attack, Adaptive Image Transformation

1 Introduction

The field of deep neural networks has developed vigorously in recent years. The models have been successfully applied to various tasks, including image classification [22,45,67], face recognition [34,50,12], semantic segmentation [3,4,5], *etc.* However, the security of the DNN models raises great concerns due to that the model is vulnerable to adversarial examples [46]. For example, an image with indistinguishable noise can mislead a well-trained classification model into the wrong category [19], or a stop sign on the road with a small elaborate patch can fool an autonomous vehicle [18]. Adversarial attack and adversarial defense are like a spear and a shield. They promote the development of each other and together improve the robustness of deep neural networks.

Our work focuses on a popular scenario in the adversarial attack, *i.e.*, transfer-based black-box attack. In this setting, the adversary can not get access to any information about the target model. Szegedy *et al.* [46] find that

Table 1: The list of image transformation methods used in various input-transformation-based adversarial attack methods

Method	Transformation	Method	Transformation
DIM [61]	Resize	CIM [63]	Crop
TIM [15]	Translate	Admix [52]	Mixup
SIM [33]	Scale	AITL (ours)	Adaptive

adversarial examples have the property of cross model transferability, *i.e.*, the adversarial example generated from a source model can also fool a target model. To further improve the transferability of adversarial examples, the subsequent works mainly adopt different input transformations [61,15,33,52] and modified gradient updates [14,33,68,63]. The former improves the transferability of adversarial examples by conducting various image transformations (*e.g.*, resizing, crop, scale, mixup) on the original images before passing through the classifier. And the latter introduces the idea of various optimizers (*e.g.*, momentum and NAG [43], Adam [28], AdaBelief [66]) into the basic iterative attack method [29] to improve the stability of the gradient and enhance the transferability of the generated adversarial examples.

Existing transfer-based attack methods have studied a variety of image transformation operations, including resizing [61], crop [63], scale [33] and so on (as listed in Tab. 1). Although effective, we find that almost all existing works of input-transformation-based methods only investigate the effectiveness of fixed image transformation operations respectively (see (a) and (b) in Fig. 1), or simply combine them in sequence (see (c) in Fig. 1) to further improve the transferability of adversarial examples. However, due to the different characteristics of each image, the most effective combination of image transformations for each image should also be different.

To solve the problem mentioned above, we propose a novel architecture called Adaptive Image Transformation Learner (AITL), which incorporates different image transformation operations into a unified framework to adaptively select the most effective combination of input transformations towards each image for improving the transferability of adversarial examples. Specifically, AITL consists of encoder and decoder models to convert discrete image transformation operations into continuous feature embeddings, as well as a predictor, which can predict the attack success rate evaluated on black-box models when incorporating the given image transformations into MIFGSM [14]. After the AITL is well-trained, we optimize the continuous feature embeddings of the image transformation through backpropagation by maximizing the attack success rate, and then use the decoder to obtain the optimized transformation operations. The adaptive combination of image transformations is used to replace the fixed combinational operations in existing methods (as shown in (d) of Fig. 1). The subsequent attack process is similar to the mainstream gradient-based attack method [29,14].

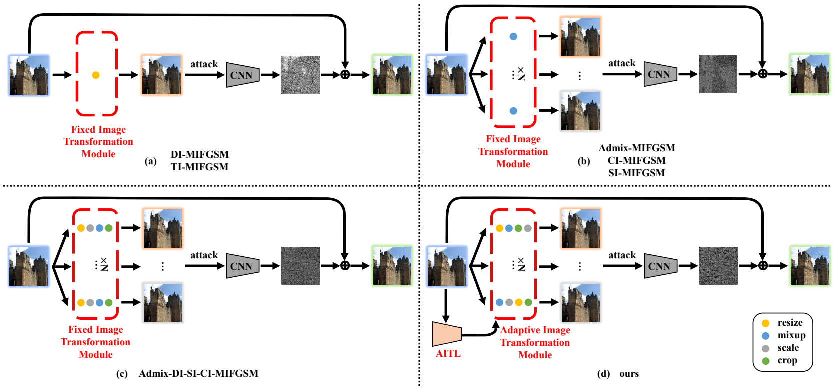
Fixed Image Transformation Module
attack
CNN
(a) DI-MIFGSM
TI-MIFGSM
×N
Fixed Image Transformation Module
(b) Admix-MIFGSM
CI-MIFGSM
SI-MIFGSM
Fixed Image Transformation Module
(c) Admix-DI-SI-CI-MIFGSM
AITL
Adaptive Image Transformation Module
(d) ours
● resize
● mixup
● scale
● crop

Fig. 1: Comparison between existing input-transformation-based black-box adversarial attack methods and our work. Different colors of the small circles in the red dotted box correspond to different image transformation operations. Existing works only conduct fixed image transformation once (as (a)) or repeat several times in parallel (as (b)), or simply combine multiple image transformation operations in the fixed sequence (as (c)). Our proposed method (as (d)) takes the characteristic of the current input image into consideration, utilizing an Adaptive Image Transformation Learner (AITL) to achieve the most effective combination of image transformations for each image, which can further improve the transferability of generated adversarial examples

Extensive experiments on ImageNet [42] demonstrate that our method not only significantly improves attack success rates on normally trained models, but also shows great effectiveness in attacking various defense models. Especially, we compare our attack method with the combination of state-of-the-art methods [14,61,33,63,52] against eleven advanced defense methods and achieve a significant improvement of 15.88% and 5.87% on average under the single model setting and the ensemble of multiple models setting, respectively. In addition, we conclude that `Scale` is the most effective operation, and geometry-based image transformations (*e.g.*, resizing, rotation, shear) can bring more improvement on the transferability of the adversarial examples, compared to other color-based image transformations (*e.g.*, brightness, sharpness, saturation).

We summarize our main contributions as follows:

1. Unlike the fixed combinational transformation used in existing works, we incorporate different image transformations into a unified framework to adaptively select the most effective combination of image transformations for the specific image.

2. We propose a novel architecture called Adaptive Image Transformation Learner (AITL), which elaborately converts discrete transformations into continuous embeddings and further adopts backpropagation to achieve the optimal solutions, *i.e.*, a combination of effective image transformations for each image.

3. We conclude that `Scale` is the most effective operation, and geometry-based image transformations are more effective than other color-based image transformations to improve the transferability of adversarial examples.

2 Related Work

2.1 Adversarial Attack

The concept of adversarial example is first proposed by Szegedy *et al.* [46]. The methods in adversarial attack can be classified as different categories according to the amount of information to the target model the adversary can access, *i.e.*, white-box attack [19,38,37,1,2,9,47,17], query-based black-box attack [6,49,24,31,7,16,36] and transfer-based black-box attack [14,61,15,56,30,21,33,51,58]. Since our work focuses on the area of transfer-based black-box attacks, we mainly introduce the methods of transfer-based black-box attack in detail.

The adversary in transfer-based black-box attack can not access any information about the target model, which only utilizes the transferability of adversarial example [19] to conduct the attack on the target model. The works in this task can be divided into two main categories, *i.e.*, modified gradient updates and input transformations.

In the branch of modified gradient updates, Dong *et al.* [14] first propose MIFGSM to stabilize the update directions with a momentum term to improve the transferability of adversarial examples. Lin *et al.* [33] propose the method of NIM, which adapts Nesterov accelerated gradient into the iterative attacks. Zou *et al.* [68] propose an Adam [28] iterative fast gradient tanh method (AI-FGTM) to generate indistinguishable adversarial examples with high transferability. Besides, Yang *et al.* [63] absorb the AdaBelief optimizer into the update of the gradient and propose ABI-FGM to further boost the success rates of adversarial examples for black-box attacks. Recently, Wang *et al.* propose the techniques of variance tuning [51] and enhanced momentum [53] to further enhance the class of iterative gradient-based attack methods.

In the branch of various input transformations, Xie *et al.* [61] propose DIM, which applies random resizing to the inputs at each iteration of I-FGSM [29] to alleviate the overfitting on white-box models. Dong *et al.* [15] propose a translation-invariant attack method, called TIM, by optimizing a perturbation over an ensemble of translated images. Lin *et al.* [33] also leverage the scale-invariant property of deep learning models to optimize the adversarial perturbations over the scale copies of the input images. Further, Crop-Invariant attack Method (CIM) is proposed by Yang *et al.* [63] to improve the transferability of adversarial. Contemporarily, inspired by mixup [65], Wang *et al.* [52] propose Admix to calculate the gradient on the input image admixed with a small portion of each add-in image while using the original label of the input, to craft more transferable adversaries. Besides, Wu *et al.* [58] propose ATTA method, which improves the robustness of synthesized adversarial examples via an adversarial transformation network. Recently, Yuan *et al.* [64] propose AutoMA to find the strong model augmentation policy by the framework of reinforcement learning. The works most relevant to ours are AutoMA [64] and ATTA [58] and we give a brief discussion on the differences between our work and theirs in Appendix B.

2.2 Adversarial Defense

To boost the robustness of neural networks and defend against adversarial attacks, numerous methods of adversarial defense have been proposed.

Adversarial training [19,29,37] adds the adversarial examples generated by several methods of adversarial attack into the training set, to boost the robustness of models. Although effective, the problems of huge computational cost and overfitting to the specific attack pattern in adversarial training receive increasing concerns. Several follow-up works [41,37,48,54,13,40,57,55] aim to solve these problems. Another major approach is the method of input transformation, which preprocesses the input to mitigate the adversarial effect ahead, including JPEG compression [20,35], denoising [32], random resizing [60], bit depth reduction [62] and so on. Certified defense [27,59,8,25] attempts to provide a guarantee that the target model can not be fooled within a small perturbation neighborhood of the clean image. Moreover, Jia *et al.* [26] utilize an image compression model to defend the adversarial examples. Naseer *et al.* [39] propose a self-supervised adversarial training mechanism in the input space to combine the benefit of both the adversarial training and input transformation method. The various defense methods mentioned above help to improve the robustness of the model.

3 Method

In this section, we first give the definition of the notations in the task. And then we introduce our proposed Adaptive Image Transformation Learner (AITL), which can adaptively select the most effective combination of image transformations used during the attack to improve the transferability of generated adversarial examples.

3.1 Notations

Let $x \in \mathcal{X}$ denote a clean image from a dataset of $\mathcal{X}$, and $y \in \mathcal{Y}$ is the corresponding ground truth label. Given a source model f with parameters θ, the objective of adversarial attack is to find the adversarial example x^{adv} that satisfies:

$$f(x^{adv}) \neq y, \quad s.t. \|x - x^{adv}\|_\infty \leq \epsilon, \tag{1}$$

where ϵ is a preset parameter to constrain the intensity of the perturbation. In implementation, most gradient-based adversaries utilize the method of maximizing the loss function to iteratively generate adversarial examples. We here take the widely used method of MIFGSM [14] as an example:

$$g_{t+1} = \mu \cdot g_t + \frac{\nabla_{x_t^{adv}} J(f(x_t^{adv}), y)}{\|\nabla_{x_t^{adv}} J(f(x_t^{adv}), y)\|_1}, \tag{2}$$

$$x_{t+1}^{adv} = x_t^{adv} + \alpha \cdot sign(g_{t+1}), \tag{3}$$

$$g_0 = 0, \quad x_0^{adv} = x, \tag{4}$$

where g_t is the accumulated gradients, x_t^{adv} is the generated adversarial example at the time step t, $J(\cdot)$ is the loss function used in classification models (*i.e.*, the cross entropy loss), μ and α are hyperparameters.

3.2 Overview of AITL

Existing works of input-transformation-based methods have studied the influence of some input transformations on the transferability of adversarial examples. These methods can be combined with the MIFGSM [14] method and can be summarized as the following paradigm, where the Eq. (2) is replaced by:

$$g_{t+1} = \mu \cdot g_t + \frac{\nabla_{x_t^{adv}} J(f(T(x_t^{adv})), y)}{\|\nabla_{x_t^{adv}} J(f(T(x_t^{adv})), y)\|_1}, \tag{5}$$

where T represents different input transformation operations in different method (*e.g.*, resizing in DIM [61], translation in TIM [15], scaling in SIM [33], cropping in CIM [63], mixup in Admix [52]).

Although existing methods improve the transferability of adversarial examples to a certain extent, almost all of these methods only utilize different image transformations respectively and haven't systematically studied which transformation operation is more suitable. Also, these methods haven't considered the characteristic of each image, but uniformly adopt a fixed transformation method for all images, which is not reasonable in nature and cannot maximize the transferability of the generated adversarial examples.

In this paper, we incorporate different image transformation operations into a unified framework and utilize an Adaptive Image Transformation Learner (AITL) to adaptively select the suitable input transformations towards different input images (as shown in (d) of Fig. 1). This unified framework can analyze the impact of different transformations on the generated adversarial examples.

Overall, our method consists of two phases, *i.e.*, the phase of training AITL to learn the relationship between various image transformations and the corresponding attack success rates, and the phase of generating adversarial examples with well-trained AITL. During the training phase, we conduct encoder and decoder networks, which can convert the discretized image transformation operations into continuous feature embeddings. In addition, a predictor is proposed to predict the attack success rate in the case of the original image being firstly transformed by the given image transformation operations and then attacked with the method of MIFGSM [14]. After the training of AITL is finished, we

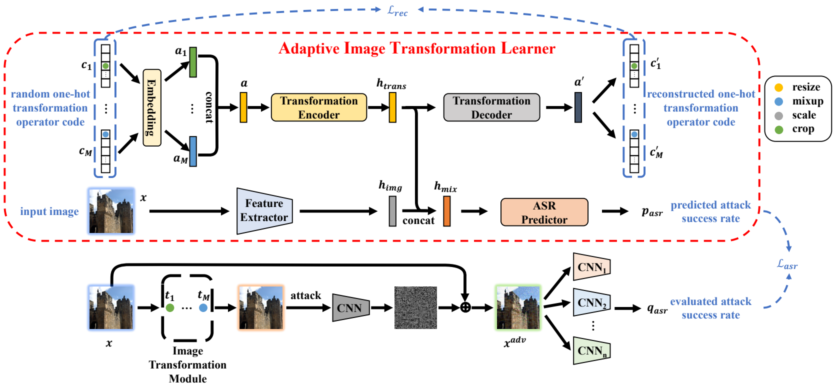

Fig. 2: The diagram of Adaptive Image Transformation Learner in the process of training

maximize the attack success rate to optimize the continuous feature embeddings of the image transformation through backpropagation, and then use the decoder to obtain the optimal transformation operations specific to the input, and incorporate it into MIFGSM to conduct the actual attack.

In the following two subsections, we will introduce the two phases mentioned above in detail, respectively.

3.3 Training AITL

The overall process of training AITL is shown in Fig. 2. We first randomly select M image transformations $t_1, t_2, \cdots, t_M$ from the image transformation operation zoo (including both geometry-based and color-based operations, for details please refer to Appendix A.3) based on uniform distribution to compose an image transformation combination. We then discretize different image transformations by encoding them into one-hot vectors $c_1, c_2, \cdots, c_M$ (*e.g.*, $[1, 0, 0, \cdots]$ represents resizing, $[0, 1, 0, \cdots]$ represents scaling). An embedding layer then converts different transformation operations into their respective feature vectors, which are concatenated into an integrated input transformation feature vector a:

$$a_1, a_2, \cdots, a_M = Embedding(c_1, c_2, \cdots, c_M), \tag{6}$$

$$a = Concat(a_1, a_2, \cdots, a_M). \tag{7}$$

The integrated input transformation feature vector then goes through a transformation encoder f_{en} and decoder f_{de} in turn, so as to learn the continuous feature embeddings h_{trans} in the intermediate layer:

$$h_{trans} = f_{en}(a), \tag{8}$$

$$a' = f_{de}(h_{trans}). \tag{9}$$

The resultant decoded feature a' is then utilized to reconstruct the input transformation one-hot vectors:

$$c'_1, c'_2, \cdots, c'_M = FC(a'), \tag{10}$$

where FC represents a fully connected layer with multiple heads, each represents the reconstruction of an input image transformation operation. On the other hand, a feature extractor f_{img} is utilized to extract the image feature of the original image h_{img}, which is concatenated with the continuous feature embeddings of image transformation combination h_{trans}:

$$h_{img} = f_{img}(x), \tag{11}$$

$$h_{mix} = Concat(h_{trans}, h_{img}). \tag{12}$$

Then the mixed feature is used to predict the attack success rate p_{asr} through an attack success rate predictor f_{pre} in the case of the original image being firstly transformed by the input image transformation combination and then attacked with the method of MIFGSM:

$$p_{asr} = f_{pre}(h_{mix}). \tag{13}$$

Loss Functions. The loss function used to train the network consists of two parts. The one is the reconstruction loss $\mathcal{L}_{rec}$ to constrain the reconstructed image transformation operations $c'_1, c'_2, \cdots, c'_M$ being consistent with the input image transformation operations $c_1, c_2, \cdots, c_M$:

$$\mathcal{L}_{rec} = -\sum_{i=1}^{M} c_i^T \log c'_i, \tag{14}$$

where T represents the transpose of a vector. The other one is the prediction loss $\mathcal{L}_{asr}$, which aims to ensure that the attack success rate predicted by the ASR predictor p_{asr} is close to the actual attack success rate q_{asr}.

$$\mathcal{L}_{asr} = \|p_{asr} - q_{asr}\|_2. \tag{15}$$

The actual attack success rate q_{asr} is achieved by evaluating the adversarial example x^{adv}, which is generated through replacing the fixed transformation operations in existing methods by the given input image transformation combination (*i.e.*, $T = t_M \circ \cdots \circ t_2 \circ t_1$ in Eq. (5)) on n black-box models $f_1, f_2, \cdots, f_n$ (as shown in the bottom half in Fig. 2):

$$q_{asr} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}(f_i(x^{adv}) \neq y). \tag{16}$$

And the total loss function is the sum of above introduced two items:

$$\mathcal{L}_{total} = \mathcal{L}_{rec} + \mathcal{L}_{asr}. \tag{17}$$

The entire training process is summarized in Algorithm 1 in the appendix.

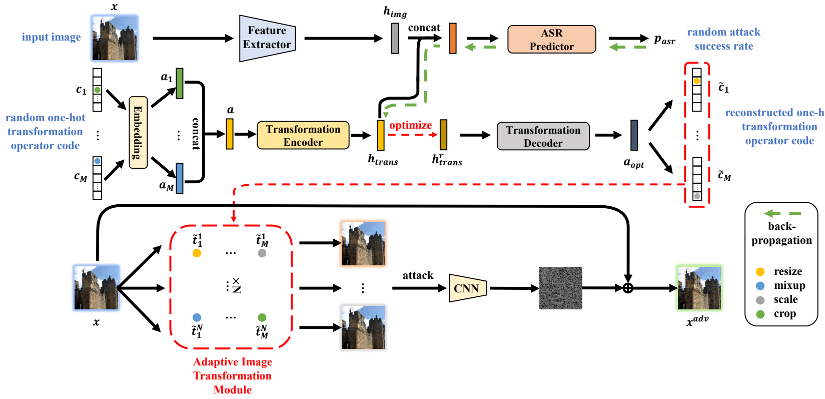

Fig. 3: The process of generating adversarial examples with Adaptive Image Transformation Learner

3.4 Generating Adversarial Examples with AITL

When the training of Adaptive Image Transformation Learner is finished, it can be used to adaptively select the appropriate combination of image transformations when conducting adversarial attacks against any unknown model. The process has been shown in Fig. 3.

For an arbitrary input image, we can not identify the most effective input transformation operations that can improve the transferability of generated adversarial examples ahead. Therefore, we first still randomly sample M initial input transformation operations $t_1, t_2, \cdots, t_M$, and go through a forward pass in AITL to get the predicted attack success rate p_{asr} corresponding to the input transformation operations. Then we iteratively optimize the image transformation feature embedding h_{trans} by maximizing the predicted attack success rate for r times:

$$h^{t+1}_{trans} = h^t_{trans} + \gamma \cdot \nabla_{h^t_{trans}} p_{asr}, \tag{18}$$

$$h^0_{trans} = h_{trans}, \tag{19}$$

where γ is the step size in each optimizing step. Finally we achieve the optimized image transformation feature embedding h^r_{trans}. Then we utilize the pre-trained decoder to convert the continuous feature embedding into specific image transformation operations:

$$a_{opt} = f_{de}(h^r_{trans}), \tag{20}$$

$$\tilde{c}_1, \tilde{c}_2, \cdots, \tilde{c}_M = FC(a_{opt}). \tag{21}$$

The resultant image transformation operations $\tilde{c}_1, \tilde{c}_2, \cdots, \tilde{c}_M$ achieved by AITL are considered to be the most effective combination of image transformations for improving the transferability of generated adversarial example towards the specific input image. Thus we utilize these image transformation operations to generate adversarial examples. When combined with MIFGSM [14], the whole process can be summarized as:

$$\tilde{c}_1, \tilde{c}_1, \cdots, \tilde{c}_M = AITL(x), \tag{22}$$

$$g_{t+1} = \mu \cdot g_t + \frac{\nabla_{x_t^{adv}} J(f(\tilde{c}_M \circ \cdots \circ \tilde{c}_1(x_t^{adv})), y)}{\|\nabla_{x_t^{adv}} J(f(\tilde{c}_M \circ \cdots \circ \tilde{c}_1(x_t^{adv})), y)\|_1}, \tag{23}$$

$$x_{t+1}^{adv} = x_t^{adv} + \alpha \cdot sign(g_{t+1}), \tag{24}$$

$$g_0 = 0, \quad x_0^{adv} = x. \tag{25}$$

Since the random image transformation operations contain randomness (*e.g.*, the degree in rotation, the width and height in resizing), existing works [33,63,52] conduct these transformation operations multiple times in parallel during each step of the attack to alleviate the impact of the instability caused by randomness on the generated adversarial examples (as shown in (b) of Fig. 1). Similar to previous works, we also randomly sample the initial image transformation combination multiple times, and then optimize them to obtain the optimal combination of image transformation operations respectively. The several optimal image transformation combinations are used in parallel to generate adversarial examples (as shown in the bottom half in Fig. 3). The entire process of using AITL to generate adversarial examples is formally summarized in Algorithm 2 in the appendix.

The specific network structure of the entire framework is shown in Appendix A.4. The ASR predictor, Transformation Encoder and Decoder in our AITL consist of only a few FC layers. During the iterative attack, our method only needs to infer once before the first iteration. So our AITL is a lightweight method, and the extra cost compared to existing methods is negligible.

4 Experiments

In this section, we first introduce the settings in the experiments in Sec. 4.1. Then we demonstrate the results of our proposed AITL method on the single model attack and an ensemble of multiple models attack, respectively in Sec. 4.2. We also analyze the effectiveness of different image transformation methods in Sec. 4.3. In Appendix C, more extra experiments are provided, including the attack success rate under different perturbation budgets, the influence of some hyperparameters, more experiments on the single model attack, the results of AITL combined with other base attack methods and the visualization of generated adversarial examples.

4.1 Settings

Dataset. We use two sets of subsets[4,5] in the ImageNet dataset [42] to conduct experiments. Each set contains 1000 images, covering almost all categories in ImageNet, which has been widely used in previous works [14,15,33]. All images have the size of $299 \times 299 \times 3$. In order to make a fair comparison with other methods, we use the former subset to train the AITL model, and evaluate all methods on the latter one.

Models. In order to avoid overfitting of the AITL model and ensure the fairness of the experimental comparison, we use completely different models to conduct experiments during the training and evaluation of the AITL model. During the training, we totally 11 models to provide the attack success rate corresponding to the input transformation, including 10 normally trained and 1 adversarially trained models. During the evaluation, we use 7 normally trained models, 3 adversarially trained models, and another 8 stronger defense models to conduct the experiments. The details are provided in Appendix A.1.

Baselines. Several input-transformation-based black-box attack methods (*e.g.*, DIM [61], TIM [15], SIM [33], CIM [63], Admix [52], AutoMA [64]) are utilized to compare with our proposed method. Unless mentioned specifically, we combine these methods with MIFGSM [14] to conduct the attack. In addition, we also combine these input-transformation-based methods together to form the strongest baseline, called Admix-DI-SI-CI-MIFGSM (as shown in (c) of Fig. 1, ADSCM for short). Moreover, we also use a random selection method instead of the AITL to choose the combination of image transformations used in the attack, which is denoted as `Random`. The details of these baselines are provided in Appendix A.2.

Image Transformation Operations. Partially referencing from [10,11], we totally select 20 image transformation operations as candidates, including `Admix`, `Scale`, `Admix-and-Scale`, `Brightness`, `Color`, `Contrast`, `Sharpness`, `Invert`, `Hue`, `Saturation`, `Gamma`, `Crop`, `Resize`, `Rotate`, `ShearX`, `ShearY`, `TranslateX`, `TranslateY`, `Reshape`, `Cutout`. The details of these operations are provided in Appendix A.3, including the accurate definitions and specific parameters in the random transformations.

Implementation Details. We train the AITL model for 10 epochs. The batch size is 64, and the learning rate β is set to 0.00005. The detailed network structure of AITL is introduced in Appendix A.4. The maximum adversarial perturbation ϵ is set to 16, with an iteration step T of 10 and step size α of 1.6. The number of iterations during optimizing image transformation features r is set to 1 and the corresponding step size γ is 15. The number of image transformation operations used in a combination M is set to 4 (the same number as the transformations used in the strongest baseline ADSCM for a fair comparison). Also, for a fair comparison of different methods, we control the number of repe-

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

[5] `https://drive.google.com/drive/folders/1CfobY6i8BfqfWPHL31FKFDipNjqWwAhS`

Table 2: Attack success rates (%) of adversarial attacks against 7 normally trained models and 11 defense models under **single model** setting. The adversarial examples are crafted on Incv3. * indicates the white-box model. † The results of AutoMA [64] are cited from their original paper

(a) The evaluation against 7 normally trained models

	Incv3*	Incv4	IncResv2	Resv2-101	Resv2-152	PNASNet	NASNet
MIFGSM [14]	**100**	52.2	50.6	37.4	35.6	42.2	42.2
DIM [61]	99.7	78.3	76.3	59.6	59.9	64.6	66.2
SIM [33]	**100**	84.5	81.3	68.0	65.3	70.8	73.6
CIM [63]	**100**	85.1	81.6	58.1	57.4	65.7	66.7
Admix [52]	99.8	69.5	66.5	55.3	55.4	60.0	62.7
ADSCM	**100**	87.9	86.1	75.8	76.0	80.9	82.2
Random	**100**	94.0	92.0	79.7	80.0	84.6	85.5
AutoMA† [64]	98.2	91.2	91.0	82.5	-	-	-
AITL (ours)	99.8	**95.8**	**94.1**	**88.8**	**90.1**	**94.1**	**94.0**
AutoMA-TIM† [64]	97.5	80.7	74.3	69.3	-	-	-
AITL-TIM (ours)	**99.8**	**93.4**	**92.1**	**91.9**	**92.2**	**93.8**	**94.6**

(b) The evaluation against 11 defense models

	Incv3$_{\text{ens3}}$	Incv3$_{\text{ens4}}$	IncResv2$_{\text{ens}}$	HGD	R&P	NIPS-r3	Bit-Red	JPEG	FD	ComDefend	RS
MIFGSM [14]	15.6	15.2	6.4	5.8	5.6	9.3	18.5	33.3	39.0	28.1	16.8
DIM [61]	31.0	29.2	13.4	15.8	14.8	24.6	26.8	59.3	45.8	48.3	21.8
SIM [33]	37.5	35.0	18.8	16.8	18.3	26.8	31.0	66.9	52.1	55.9	24.1
CIM [63]	33.3	30.0	15.9	20.4	16.4	25.7	26.8	62.2	46.3	44.9	21.2
Admix [52]	27.5	27.0	14.3	11.6	12.6	19.8	28.4	51.2	48.8	44.0	22.0
ADSCM	49.3	46.9	27.0	33.1	28.5	40.5	39.0	73.0	60.4	65.5	32.8
Random	49.8	46.7	24.5	29.2	26.4	42.2	36.3	81.4	57.4	69.6	29.6
AutoMA† [64]	49.2	49.0	29.1	-	-	-	-	-	-	-	-
AITL (ours)	**69.9**	**65.8**	**43.4**	**50.4**	**46.9**	**59.9**	**51.6**	**87.1**	**73.0**	**83.2**	**39.5**
AutoMA-TIM† [64]	74.8	74.3	63.6	65.7	62.9	68.1	-	-	**84.7**	-	-
AITL-TIM (ours)	**81.3**	**78.9**	**69.1**	**75.1**	**64.7**	**74.6**	**60.9**	**87.8**	83.8	**85.6**	**55.4**

titions per iteration in all methods to 5 (m in SIM [33], m_2 in Admix [52], m in AutoMA [64] and N in our AITL).

4.2 Compared with the State-of-the-art Methods

Attack on the Single Model. We use Inceptionv3 [45] model as the white-box model to conduct the adversarial attack, and evaluate the generated adversarial examples on both normally trained models and defense models. As shown in Tab. 2, comparing various existing input-transformation-based methods, our proposed AITL significantly improves the attack success rates against various black-box models. Especially for the defense models, although it is relatively difficult to attack successfully, our method still achieves a significant improvement of 15.88% on average, compared to the strong baseline (Admix-DI-SI-CI-MIFGSM). It demonstrates that, compared with the fixed image transformation combination, adaptively selecting combinational image transformations for each image can indeed improve the transferability of adversarial examples. Also, when compared to AutoMA [64], our AITL achieves a distinct improvement, which shows that our AITL model achieves better mapping between discrete image

Table 3: Attack success rates (%) of adversarial attacks against 7 normally trained models and 11 defense models under **multiple models** setting. The adversarial examples are crafted on the ensemble of Incv3, Incv4, IncResv2 and Resv2-101. * indicates the white-box model. † The results of AutoMA [64] are cited from their original paper

(a) The evaluation against 7 normally trained models

	Incv3*	Incv4*	IncResv2*	Resv2-101*	Resv2-152	PNASNet	NASNet
MIFGSM [14]	**100**	99.6	99.7	**98.5**	86.8	79.4	81.2
DIM [61]	99.5	99.4	98.9	96.9	92.0	91.3	92.1
SIM [33]	99.9	99.1	98.3	93.2	91.7	90.9	91.9
CIM [63]	99.8	99.3	97.8	90.6	88.5	88.2	90.9
Admix [52]	99.9	99.5	98.2	95.4	89.3	88.1	90.0
ADSCM	99.8	99.3	99.2	96.9	96.0	88.1	90.0
Random	**100**	99.4	98.9	96.9	94.3	94.4	95.0
AITL (ours)	99.9	**99.7**	**99.9**	97.3	**96.6**	**97.7**	**97.8**

(b) The evaluation against 11 defense models

	$\text{Incv3}_{\text{ens3}}$	$\text{Incv3}_{\text{ens4}}$	$\text{IncResv2}_{\text{ens}}$	HGD	R&P	NIPS-r3	Bit-Red	JPEG	FD	ComDefend	RS
MIFGSM [14]	52.4	47.5	30.1	39.2	31.7	43.6	33.8	76.4	54.5	66.8	29.7
DIM [61]	77.4	73.1	54.4	68.4	61.2	73.5	53.3	89.8	71.5	84.3	43.1
SIM [33]	78.8	74.4	59.8	66.9	59.0	70.7	58.1	89.0	73.2	83.0	46.6
CIM [63]	75.1	69.7	54.3	68.5	59.1	70.7	51.1	90.2	68.9	78.5	41.1
Admix [52]	67.7	61.9	44.8	51.0	44.8	57.9	51.4	84.6	69.2	78.5	42.2
ADSCM	85.8	82.9	69.2	78.7	74.1	81.1	68.1	94.9	82.3	90.8	57.8
Random	83.7	80.2	64.8	73.7	67.3	77.9	65.7	93.0	79.9	88.6	52.3
AITL (ours)	**89.3**	**89.0**	**79.0**	**85.5**	**82.3**	**86.3**	**74.9**	**96.2**	**88.4**	**93.7**	**65.7**
AutoMA-TIM† [64]	93.0	93.2	90.7	91.2	90.4	92.0	-	-	94.1	-	-
AITL-TIM (ours)	**93.8**	**95.3**	**92.0**	**93.1**	**93.7**	**94.8**	**80.9**	**95.0**	**96.2**	**95.0**	**76.9**

transformations and continuous feature embeddings. More results of attacking other models are available in Appendix C.2. Noting that the models used for evaluation here are totally different from the models used when training the AITL, our method shows great cross model transferability to conduct the successful adversarial attack.

Attack on the Ensemble of Multiple Models. We use the ensemble of four models, *i.e.*, Inceptionv3 [45], Inceptionv4 [44], Inception-ResNetv2 [44] and ResNetv2-101 [23], as the white-box models to conduct the adversarial attack. As shown in Tab. 3, compared with the fixed image transformation method, our AITL significantly improves the attack success rates on various models. Although the strong baseline ADSCM has achieved relatively high attack success rates, our AITL still obtains an improvement of 1.44% and 5.87% on average against black-box normally trained models and defense models, respectively. Compared to AutoMA [64], our AITL also achieves higher attack success rates on defense models, which shows the superiority of our proposed novel architecture.

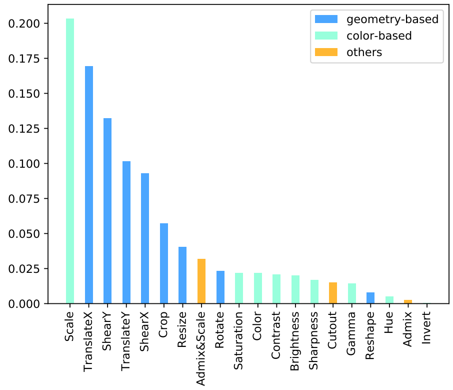

Fig. 4: The frequency of various image transformations used in AITL when generating adversarial examples of the 1000 images in ImageNet

4.3 Analysis on Image Transformation Operations

In order to further explore the effects of different image transformation operations on improving the transferability of adversarial examples, we calculate the frequency of various image transformations used in AITL when generating adversarial examples of the 1000 images in ImageNet. From Fig. 4, we can clearly see that `Scale` operation is the most effective method within all 20 candidates. Also, we conclude that the geometry-based image transformations are more effective than other color-based image transformations to improve the transferability of adversarial examples.

5 Conclusion

In our work, unlike the fixed image transformation operations used in almost all existing works of transfer-based black-box attack, we propose a novel architecture, called Adaptive Image Transformation Learner (AITL), which incorporates different image transformation operations into a unified framework to further improve the transferability of adversarial examples. By taking the characteristic of each image into consideration, our designed AITL adaptively selects the most effective combination of image transformations for the specific image. Extensive experiments on ImageNet demonstrate that our method significantly improves the attack success rates both on normally trained models and defense models under different settings.

Acknowledgments. This work is partially supported by National Key R&D Program of China (No. 2017YFA0700800), National Natural Science Foundation of China (Nos. 62176251 and Nos. 61976219).

References

1. Athalye, A., Carlini, N., Wagner, D.A.: Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In: ICML. vol. 80, pp. 274–283 (2018)
2. Athalye, A., Engstrom, L., Ilyas, A., Kwok, K.: Synthesizing robust adversarial examples. In: ICML. vol. 80, pp. 284–293 (2018)
3. Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic image segmentation with deep convolutional nets and fully connected crfs. In: ICLR (2015)
4. Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE TPAMI **40**(4), 834–848 (2018)
5. Chen, L., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587 (2017)
6. Chen, P., Zhang, H., Sharma, Y., Yi, J., Hsieh, C.: 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, AISec@CCS 2017, Dallas, TX, USA, November 3, 2017. pp. 15–26 (2017)
7. Cheng, S., Dong, Y., Pang, T., Su, H., Zhu, J.: Improving black-box adversarial attacks with a transfer-based prior. In: NeurIPS. pp. 10932–10942 (2019)
8. Croce, F., Hein, M.: Provable robustness against all adversarial \$l_p\$-perturbations for \$p\geq 1\$. In: ICLR (2020)
9. Croce, F., Hein, M.: Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In: ICML. vol. 119, pp. 2206–2216 (2020)
10. Cubuk, E.D., Zoph, B., Mané, D., Vasudevan, V., Le, Q.V.: Autoaugment: Learning augmentation strategies from data. In: CVPR. pp. 113–123 (2019)
11. Cubuk, E.D., Zoph, B., Shlens, J., Le, Q.: Randaugment: Practical automated data augmentation with a reduced search space. In: NeurIPS (2020)
12. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. In: CVPR. pp. 4690–4699 (2019)
13. Dong, Y., Deng, Z., Pang, T., Zhu, J., Su, H.: Adversarial distributional training for robust deep learning. In: NeurIPS (2020)
14. Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., Li, J.: Boosting adversarial attacks with momentum. In: CVPR. pp. 9185–9193 (2018)
15. Dong, Y., Pang, T., Su, H., Zhu, J.: Evading defenses to transferable adversarial examples by translation-invariant attacks. In: CVPR. pp. 4312–4321 (2019)
16. Du, J., Zhang, H., Zhou, J.T., Yang, Y., Feng, J.: Query-efficient meta attack to deep neural networks. In: ICLR (2020)
17. Duan, R., Chen, Y., Niu, D., Yang, Y., Qin, A.K., He, Y.: Advdrop: Adversarial attack to dnns by dropping information. In: ICCV. pp. 7506–7515 (2021)
18. Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., Prakash, A., Kohno, T., Song, D.: Robust physical-world attacks on deep learning visual classification. In: CVPR. pp. 1625–1634 (2018)
19. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: ICLR (2015)
20. Guo, C., Rana, M., Cissé, M., van der Maaten, L.: Countering adversarial images using input transformations. In: ICLR (2018)
21. Guo, Y., Li, Q., Chen, H.: Backpropagating linearly improves transferability of adversarial examples. In: NeurIPS (2020)

22. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016)
23. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: ECCV. pp. 630–645 (2016)
24. Ilyas, A., Engstrom, L., Athalye, A., Lin, J.: Black-box adversarial attacks with limited queries and information. In: ICML. vol. 80, pp. 2142–2151 (2018)
25. Jia, J., Cao, X., Wang, B., Gong, N.Z.: Certified robustness for top-k predictions against adversarial perturbations via randomized smoothing. In: ICLR (2020)
26. Jia, X., Wei, X., Cao, X., Foroosh, H.: Comdefend: An efficient image compression model to defend adversarial examples. In: CVPR. pp. 6084–6092 (2019)
27. Katz, G., Barrett, C.W., Dill, D.L., Julian, K., Kochenderfer, M.J.: Reluplex: An efficient SMT solver for verifying deep neural networks. In: CAV. vol. 10426, pp. 97–117 (2017)
28. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015)
29. Kurakin, A., Goodfellow, I.J., Bengio, S.: Adversarial machine learning at scale. In: ICLR (2017)
30. Li, M., Deng, C., Li, T., Yan, J., Gao, X., Huang, H.: Towards transferable targeted attack. In: CVPR. pp. 638–646 (2020)
31. Li, Y., Li, L., Wang, L., Zhang, T., Gong, B.: NATTACK: learning the distributions of adversarial examples for an improved black-box attack on deep neural networks. In: ICML. vol. 97, pp. 3866–3876 (2019)
32. Liao, F., Liang, M., Dong, Y., Pang, T., Hu, X., Zhu, J.: Defense against adversarial attacks using high-level representation guided denoiser. In: CVPR. pp. 1778–1787 (2018)
33. Lin, J., Song, C., He, K., Wang, L., Hopcroft, J.E.: Nesterov accelerated gradient and scale invariance for adversarial attacks. In: ICLR (2020)
34. Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., Song, L.: Sphereface: Deep hypersphere embedding for face recognition. In: CVPR. pp. 6738–6746 (2017)
35. Liu, Z., Liu, Q., Liu, T., Xu, N., Lin, X., Wang, Y., Wen, W.: Feature distillation: Dnn-oriented JPEG compression against adversarial examples. In: CVPR. pp. 860–868 (2019)
36. Ma, C., Chen, L., Yong, J.: Simulating unknown target models for query-efficient black-box attacks. In: CVPR. pp. 11835–11844 (2021)
37. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: ICLR (2018)
38. Moosavi-Dezfooli, S., Fawzi, A., Frossard, P.: Deepfool: A simple and accurate method to fool deep neural networks. In: CVPR. pp. 2574–2582 (2016)
39. Naseer, M., Khan, S.H., Hayat, M., Khan, F.S., Porikli, F.: A self-supervised approach for adversarial robustness. In: CVPR. pp. 259–268 (2020)
40. Pang, T., Yang, X., Dong, Y., Xu, T., Zhu, J., Su, H.: Boosting adversarial training with hypersphere embedding. In: NeurIPS (2020)
41. Rozsa, A., Rudd, E.M., Boult, T.E.: Adversarial diversity and hard positive generation. In: CVPRW. pp. 410–417 (2016)
42. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M.S., Berg, A.C., Fei-Fei, L.: Imagenet large scale visual recognition challenge. IJCV **115**(3), 211–252 (2015)
43. Sutskever, I., Martens, J., Dahl, G.E., Hinton, G.E.: On the importance of initialization and momentum in deep learning. In: ICML. vol. 28, pp. 1139–1147 (2013)
44. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: AAAI. pp. 4278–4284 (2017)

45. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: CVPR. pp. 2818–2826 (2016)
46. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I.J., Fergus, R.: Intriguing properties of neural networks. In: ICLR (2014)
47. Tramèr, F., Carlini, N., Brendel, W., Madry, A.: On adaptive attacks to adversarial example defenses. In: NeurIPS (2020)
48. Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I.J., Boneh, D., McDaniel, P.D.: Ensemble adversarial training: Attacks and defenses. In: ICLR (2018)
49. Uesato, J., O'Donoghue, B., Kohli, P., van den Oord, A.: Adversarial risk and the dangers of evaluating against weak attacks. In: ICML. vol. 80, pp. 5032–5041 (2018)
50. Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., Li, Z., Liu, W.: Cosface: Large margin cosine loss for deep face recognition. In: CVPR. pp. 5265–5274 (2018)
51. Wang, X., He, K.: Enhancing the transferability of adversarial attacks through variance tuning. In: CVPR. pp. 1924–1933 (2021)
52. Wang, X., He, X., Wang, J., He, K.: Admix: Enhancing the transferability of adversarial attacks. arXiv preprint arXiv:2102.00436 (2021)
53. Wang, X., Lin, J., Hu, H., Wang, J., He, K.: Boosting adversarial transferability through enhanced momentum. arXiv preprint arXiv:2103.10609 (2021)
54. Wang, Y., Ma, X., Bailey, J., Yi, J., Zhou, B., Gu, Q.: On the convergence and robustness of adversarial training. In: ICML. vol. 97, pp. 6586–6595 (2019)
55. Wong, E., Rice, L., Kolter, J.Z.: Fast is better than free: Revisiting adversarial training. In: ICLR (2020)
56. Wu, D., Wang, Y., Xia, S., Bailey, J., Ma, X.: Skip connections matter: On the transferability of adversarial examples generated with resnets. In: ICLR (2020)
57. Wu, D., Xia, S., Wang, Y.: Adversarial weight perturbation helps robust generalization. In: NeurIPS (2020)
58. Wu, W., Su, Y., Lyu, M.R., King, I.: Improving the transferability of adversarial samples with adversarial transformations. In: CVPR. pp. 9024–9033 (2021)
59. Xiao, K.Y., Tjeng, V., Shafiullah, N.M.M., Madry, A.: Training for faster adversarial robustness verification via inducing relu stability. In: ICLR (2019)
60. Xie, C., Wang, J., Zhang, Z., Ren, Z., Yuille, A.L.: Mitigating adversarial effects through randomization. In: ICLR (2018)
61. Xie, C., Zhang, Z., Zhou, Y., Bai, S., Wang, J., Ren, Z., Yuille, A.L.: Improving transferability of adversarial examples with input diversity. In: CVPR. pp. 2730–2739 (2019)
62. Xu, W., Evans, D., Qi, Y.: Feature squeezing: Detecting adversarial examples in deep neural networks. In: NDSS (2018)
63. Yang, B., Zhang, H., Zhang, Y., Xu, K., Wang, J.: Adversarial example generation with adabelief optimizer and crop invariance. arXiv preprint arXiv:2102.03726 (2021)
64. Yuan, H., Chu, Q., Zhu, F., Zhao, R., Liu, B., Yu, N.H.: Automa: Towards automatic model augmentation for transferable adversarial attacks. IEEE TMM (2021)
65. Zhang, H., Cissé, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. In: ICLR (2018)
66. Zhuang, J., Tang, T., Ding, Y., Tatikonda, S.C., Dvornek, N.C., Papademetris, X., Duncan, J.S.: Adabelief optimizer: Adapting stepsizes by the belief in observed gradients. In: NeurIPS (2020)
67. Zoph, B., Vasudevan, V., Shlens, J., Le, Q.V.: Learning transferable architectures for scalable image recognition. In: CVPR. pp. 8697–8710 (2018)

68. Zou, J., Pan, Z., Qiu, J., Duan, Y., Liu, X., Pan, Y.: Making adversarial examples more transferable and indistinguishable. arXiv preprint arXiv:2007.03838 (2020)