This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Meta Gradient Adversarial Attack

Zheng Yuan[1,2], Jie Zhang[1,2], Yunpei Jia[1,2], Chuanqi Tan[3], Tao Xue[3], Shiguang Shan[1,2]
[1]Institute of Computing Technology, Chinese Academy of Sciences
[2]University of Chinese Academy of Sciences [3] Tencent
{zheng.yuan, yunpei.jia}@vipl.ict.ac.cn; {zhangjie,sgshan}@ict.ac.cn;
{jamestan,emmaxue}@tencent.com

Abstract

In recent years, research on adversarial attacks has become a hot spot. Although current literature on the transfer-based adversarial attack has achieved promising results for improving the transferability to unseen black-box models, it still leaves a long way to go. Inspired by the idea of meta-learning, this paper proposes a novel architecture called Meta Gradient Adversarial Attack (MGAA), which is plug-and-play and can be integrated with any existing gradient-based attack method for improving the cross-model transferability. Specifically, we randomly sample multiple models from a model zoo to compose different tasks and iteratively simulate a white-box attack and a black-box attack in each task. By narrowing the gap between the gradient directions in white-box and black-box attacks, the transferability of adversarial examples on the black-box setting can be improved. Extensive experiments on the CIFAR10 and ImageNet datasets show that our architecture outperforms the state-of-the-art methods for both black-box and white-box attack settings.

1. Introduction

With the rapid development of neural networks in recent years, the reliability of neural networks has gradually attracted more and more attention. The neural networks are exceedingly sensitive to adversarial examples, *i.e.*, the imperceptible perturbation on the input can easily fool the model, leading to unexpected mistakes. For example, when employing face recognition technology for payment, a slight perturbation on the face image may trick the face recognition model into recognizing as someone else. Since attack and defense are two complementary aspects, the researches on adversarial attacks can ultimately improve the robustness of the model, thereby making the model more reliable.

In recent years, many methods have been proposed to improve the success rate of attacks against white-box models, such as FGSM [15], C&W [2], PGD [31], BIM [21], DeepFool [32], *etc*. Based on the access to model parameters, these methods can make the model misclassify the input images by only adding human-imperceptible perturbations, which is named as the white-box attack. However, in reality, a more practical scenario is that the attacker cannot obtain any information of the target model, that is, the black-box attack. Therefore, some methods turn to utilize the transferability of adversarial examples to conduct black-box attacks, such as MIM [8], DIM [50], TIM [9], NI-SI [27], *etc*. Although most of these methods have achieved promising results under the scenario of black-box attacks, the transferability of adversarial examples is still limited due to the discrepancy between the white-box models and unseen black-box models.

Inspired by the philosophy of meta-learning, we propose a novel architecture named Meta Gradient Adversarial Attack (MGAA), which is plug-and-play and can be incorporated with any gradient-based adversarial attack method. The main idea of MGAA is to generate the adversarial examples by iteratively simulating white-box and black-box attacks to improve the transferability. Specifically, as shown in Fig. 1, in each iteration, multiple models are randomly sampled from the model zoo to compose a task, which can be divided into the meta-train and the meta-test step. The meta-train step first uses an ensemble of multiple models to simulate white-box attacks to obtain temporary adversarial examples, which are then used as a basis by the meta-test step to obtain the perturbation by simulating a black-box attack scenario. Finally, the perturbation obtained during the meta-test step is added to the adversarial examples generated in the previous iteration. In Section 3.3, more theoretical analyses demonstrate that our proposed architecture can gradually narrow the gap of gradient directions between white-box attack and black-box attack settings, thus improving the transferability of generated adversarial examples. Different from vanilla meta-learning methods, which enhance the generalization through model training, our proposed MGAA directly utilizes the gradient informa-

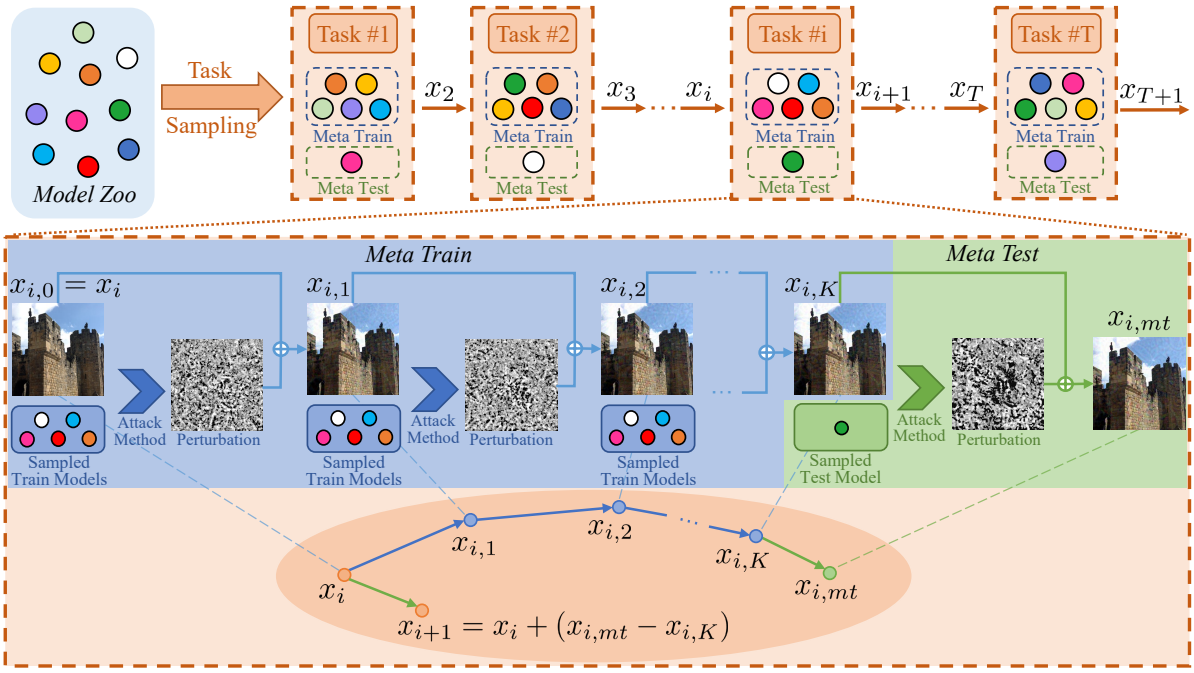

Figure 1: Overview of our Meta Gradient Adversarial Attack (MGAA). MGAA consists of multiple (T in the figure) iterations. In each iteration, $n+1$ models are randomly sampled from the model zoo to compose a meta-task. Each task is divided into two steps: the meta-train step and the meta-test step. In the meta-train step, an ensemble of the n sampled models is utilized to conduct the gradient-based attack to generate perturbations, which can be repeated by K times. The meta-test step uses the adversarial example $x_{i,K}$ obtained from the meta-train step as the basis, and utilizes the last sampled model to generate perturbations for adversarial attacks. Finally, the perturbation generated in the meta-test step $x_{i,mt} - x_{i,K}$ is added to x_i, the final adversarial example after the i-th task.

tion to improve the transferability of the adversarial examples without the need of training an extra model.

Extensive experiments on the CIFAR10 [20] and ImageNet [38] dataset demonstrate that our proposed architecture significantly improves the success rates of both white-box and black-box attacks. Especially, by integrating TI-DIM [9] method into our proposed architecture, the average attack success rate under the targeted attack setting against 10 white-box models increases by 27.67%, and the attack success rate against 6 black-box models increases by 28.52% on ImageNet, which clearly shows the superiority of our method.

The main contributions of this paper are as follows:

1. We propose the Meta Gradient Adversarial Attack architecture inspired by the philosophy of meta-learning, which iteratively simulates the white-box and the black-box attack to narrow the gap of the gradient directions when generating adversarial examples, thereby further improving the transferability of adversarial examples.

2. The proposed architecture can be combined with any existing gradient-based attack method in a plug-and-play mode.

3. Extensive experiments show that the proposed architecture can significantly improve the attack success rates under both white-box and black-box settings.

2. Related Work

In this section, we will give a brief introduction to the related works, *i.e.*, the adversarial attack, the adversarial defense, and the meta-learning.

2.1. Adversarial Attack

The task of adversarial attack is generally classified into four categories according to the amount of target model information we can access: white-box attack, score-based black-box attack, decision-based black-box attack, and transfer-based black-box attack.

White-box Attack. The white-box attack can obtain all the information of the target model, including model parameters, model structure, gradient, *etc*. FGSM [15] utilizes gradient information to update the adversarial example in one step along the direction of maximum classification loss. This method is extended by BIM [21] to propose an iterative method to generate adversarial examples through multi-step updates. PGD [31] is similar to BIM, except that it randomly selects an initial point in the neighborhood of the benign example as the starting point of the iterative attack. Some methods consider the task from the perspective of optimization. In DeepFool [32], an optimized method is employed to generate the smallest perturbation while meet-

ing the target of a successful attack. C&W [2] transforms the task into a constrained optimization problem and compares the effects of multiple objective functions.

Score-based Black-box Attack. This category of attack methods assumes they can obtain the classification probabilities of a given input image from the target model. ZOO [3] proposes a zeroth-order optimization-based method, which directly estimates the gradients of the target model for generating adversarial examples. However, it suffers from a low attack success rate and poor query efficiency since it is non-trivial to estimate the gradient with limited information. To address these problems, P-RGR [5] utilizes the transfer-based prior and query information to improve query efficiency. $\mathcal{N}$ ATTACK [25] further uses a probability density distribution over a small region centered around the input to generate adversarial examples, which defeats both vanilla DNNs and those generated by various defense techniques developed.

Decision-based Black-box Attack. Under the decision-based black-box attack setting, only the predicted class of a given input image from the target model is available, which seems more difficult than the score-based black-box attack. Boundary Attack [1] is firstly proposed to this problem, which gradually minimizes the adversarial perturbation by the approach of random walking while maintaining the aggressiveness of the adversarial example. Cheng *et al*. [4] transforms the problem into a continuous real-valued optimization problem that can be solved by any zeroth-order optimization algorithm. Different from previous methods, Dong *et al*. [10] proposes an evolutionary attack algorithm, which reduces the dimension of the search space and improves the query efficiency.

Transfer-based Black-box Attack. The transfer-based black-box attack cannot obtain any information of the target model, which is the most challenging setting. Existing methods [8, 50, 9, 54, 27] mainly endeavor to improve the attack success rates by the transferability of the adversarial examples generated from the substitute model. We will review these methods in Sec. 3.1 in details.

Although these methods have been proposed to gradually improve the transferability of adversarial examples, the results in black-box attacks still leave a lot of room for improvement. Our work solves this problem from the perspective of meta-learning, which reduces the gap between white-box and black-box attacks, thus achieving better results.

2.2. Adversarial Defense

The task of adversarial defense is to improve the robustness of the model so that the model can correctly classify the perturbed adversarial examples. Defense methods can be divided into five categories: adversarial training, input transformation, randomization, model ensemble, and certified defenses. Adversarial training [45, 31, 41] conducts the model training based on the generated adversarial examples. Input transformation utilizes JPEG compression [12], denoising [26], and extra GAN model [39] before feeding the images to the model. Randomization refers to adding random noise into the input examples [49] or models [7] to make the model more robust to adversarial examples. The model ensemble means the ensemble of multiple models in the output layer. Compared with a single model, it can reduce the impact of distributions in the adversarial examples [22, 29]. In addition, some work [37, 46] proves that under a specific target model, a certified defense model can ensure robustness to adversarial examples.

2.3. Meta-learning

Meta-learning is a concept of learning how to learn. It mainly focuses on how to use a large amount of data to learn a learning mode. When encountering a new task later, only a small amount of additional data is needed to quickly complete the new task by fine-tuning the model [13, 34]. As a newly proposed method in recent years, meta-learning has been widely used in various tasks. Meta-Attack [11] uses the meta-learning framework to train the gradient estimator. In the case of the score-based black-box attack, only a small number of queries are needed to fine-tune the gradient estimator to complete the adversarial attack.

Different from Meta-Attack, our proposed MGAA is actually not a meta-learning framework but only inspired by its philosophy. Specifically, we do not need to explicitly train an extra model but directly generate the adversarial examples in the process of iteration.

3. Methodology

Since our proposed architecture is plug-and-play and can be integrated with any existing gradient-based attack method, we first introduce some typical gradient-based attack methods in Sec. 3.1. And then, we introduce our proposed Meta Gradient Adversarial Attack architecture in detail in Sec. 3.2. Moreover, we give a detailed analysis of the effectiveness of our proposed architecture theoretically in Sec. 3.3. Finally, a discussion on the differences between our proposed MGAA and some related works is given in Sec. 3.4.

Let x denote the benign example and y denote the true label corresponding to the benign example. $f(x) : \mathcal{X} \to \mathcal{Y}$ denotes the target classifier, J denotes the loss function used for training the classifier, which usually means cross entropy loss. x^{adv} denotes the adversarial example that needs to be optimized. The optimization objective can be described as:

$$\arg\max_{x^{adv}} J(f(x^{adv}), y), \quad \text{s.t.} \|x^{adv} - x\|_\infty \leq \epsilon, \qquad (1)$$

where ϵ is the maximum perturbation per pixel allowed to

add in the benign example. The goal of the adversarial example is to mislead the classifier into misclassification ($f(x^{adv} \neq y)$) with less perturbation added in compared to the benign example.

3.1. Gradient-Based Attack Method

We briefly introduce several gradient-based attack methods in this section, which focus on solving the problem of the white-box or transfer-based black-box attack.

Fast Gradient Sign Method. FGSM [15] is a one-step method for white-box attack, which update the adversarial example in the direction of maximizing the loss function:

$$x^{adv} = x + \epsilon \cdot \text{sign}(\nabla_x J(f(x), y)). \tag{2}$$

Basic Iterative Method. Extended by FGSM, BIM [21] proposes an iteration method to improve the success rate of white-box attack, which can be described as:

$$x_{t+1}^{adv} = x_t^{adv} + \alpha \cdot \text{sign}(\nabla_x J(f(x_t^{adv}), y)), \tag{3}$$

where $x_0^{adv} = x$, $\alpha = \epsilon/T$ and T is the number of iterations.

Momentum Iterative Fast Gradient Sign Method. A momentum term is further proposed in MIM [8] to accumulate the update direction of previous steps in the process of iteration:

$$g_{t+1} = \mu \cdot g_t + \frac{\nabla_x J(f(x_t^{adv}), y)}{\|\nabla_x J(f(x_t^{adv}), y)\|_1}, \tag{4}$$

$$x_{t+1}^{adv} = x_t^{adv} + \alpha \cdot \text{sign}(g_{t+1}), \tag{5}$$

where g_t denotes the momentum term of gradient in the t-th iteration and μ is a decay factor. Compared with BIM, MIM achieves significant progress in the black-box attack.

Diverse Inputs Method. DIM [50] proposes to apply random resizing and padding transformations in the benign example to improve the transferability of adversarial examples. Although this method is simple and easy to implement, it also brings great improvements.

Translation-Invariant Attack Method. Different from previous methods, TIM [9] proposes to convolve the gradient with a Gaussian kernel, which can be combined with MIM:

$$g_{t+1} = \mu \cdot g_t + \frac{W * \nabla_x J(f(x_t^{adv}), y)}{\|W * \nabla_x J(f(x_t^{adv}), y)\|_1}, \tag{6}$$

$$x_{t+1}^{adv} = x_t^{adv} + \alpha \cdot \text{sign}(g_{t+1}), \tag{7}$$

where W is a Gaussian kernel and $*$ denotes the operator of convolution.

SI-NI. Moreover, SI-NI [27] proposes two approaches to further improve the transferability of adversarial examples, *i.e.*, NI-FGSM and SIM. NI-FGSM integrates Nesterov Accelerated Gradient [33] into the iterative gradient-based attack, *e.g.* MIM, to conduct a robust adversarial attack:

$$x_t^{nes} = x_t^{adv} + \alpha \cdot \mu \cdot g_t, \tag{8}$$

$$g_{t+1} = \mu \cdot g_t + \frac{\nabla_x J(f(x_t^{nes}), y)}{\|\nabla_x J(f(x_t^{nes}), y)\|_1}, \tag{9}$$

$$x_{t+1}^{adv} = x_t^{adv} + \alpha \cdot \text{sign}(g_{t+1}). \tag{10}$$

SIM optimizes the adversarial examples over the scale copies of the benign example:

$$\arg\max_{x^{adv}} \frac{1}{m} \sum\nolimits_{i=0}^{m} J(f(S_i(x^{adv})), y), \quad \text{s.t.} \|x^{adv} - x\|_\infty \leq \epsilon, \tag{11}$$

where $S_i(x) = x/2^i$ denotes the scaled image and m denotes the number of scale copies.

3.2. Meta Gradient Adversarial Attack

Most of the existing transfer-based black-box attack methods use gradient-based methods to generate adversarial examples against the white-box models, and then directly utilize the transferability of the adversarial examples to conduct the attack against the black-box model. However, due to the differences in the structures and the parameters between different models, relying only on the transferability of adversarial examples to conduct the black-box attack cannot achieve perfect results, which leaves a lot of room for improvement.

In this paper, we solve the problem of the black-box attack from a different perspective. Inspired by the philosophy of meta-learning, we propose an architecture called Meta Gradient Adversarial Attack (MGAA), which is plug-and-play and can be integrated with any existing gradient-based attack method. As shown in Fig. 1, T tasks are sampled from a model zoo iteratively. In each task, we simulate a white-box attack in the meta-train step and a black-box attack in the meta-test step, narrowing the gap of gradient directions between white-box and black-box settings to improve the transferability.

Specifically, suppose there are a total of N models of $M_1, M_2, \cdots, M_N$ in the model zoo, we randomly select $n+1$ models to compose a task in each iteration, which consists of two steps, *i.e.*, meta-train and meta-test. At the beginning of each iteration (take i-th iteration as an example), we take the adversarial example generated in the previous iteration x_i as the input, denoted as $x_{i,0}$.

Meta-train. A total of n models of $M_{k_1}, M_{k_2}, \cdots, M_{k_n}$ are utilized to simulate the white-box attack. We employ the same approach as [8] when doing the adversarial attack by the ensemble of multiple models. Specifically, to attack the ensemble of n models, we fuse the logits as:

$$l(x_{i,0}) = \sum\nolimits_{s=1}^{n} w_s l_{k_s}(x_{i,0}), \tag{12}$$

where $l_{k_s}(x_{i,0})$ is the logits of the model M_{k_s}, w_s is the ensemble weight of each model with $w_s \geq 0$ and $\sum_{s=1}^{n} w_s = 1$. Then the cross entropy loss is used to calculate the loss of misclassification:

$$\mathcal{L}_{M_{k_1},\cdots,M_{k_n}}(x_{i,0}) = -\mathbb{1}_y \cdot \log(\text{softmax}(l(x_{i,0}))), \quad (13)$$

where $\mathbb{1}_y$ is the one-hot encoding of y. Same as the common method of gradient-based attack, the adversarial example is updated along the direction of maximizing the loss function:

$$x_{i,j+1} = x_{i,j} + \alpha \cdot \text{sign}(\nabla_{x_{i,j}} \mathcal{L}_{M_{k_1},\cdots,M_{k_n}}(x_{i,j})), \quad (14)$$

where α is the step size in meta-train step. The meta-train step can be iterated for K times, and the subscript j denotes the number of iteration in meta-train step. It is worth noting that our proposed architecture can be integrated with any gradient-based attack methods, and we only take the formula of update in BIM [21] as an example here for convenience.

Meta-test. After using the ensemble of multiple models to simulate the white-box attack, we use the last sampled model $M_{k_{n+1}}$ to simulate the black-box attack on the basis of generated adversarial example $x_{i,K}$ in the meta-train step, where K is the number of iteration in meta-train step.

Specifically, we first calculate the cross entropy loss by model $M_{k_{n+1}}$:

$$\mathcal{L}_{M_{k_{n+1}}}(x_{i,K}) = -\mathbb{1}_y \cdot \log(\text{softmax}(l_{k_{n+1}}(x_{i,K}))), \quad (15)$$

where $l_{k_{n+1}}(x_{i,K})$ is the logits of model $M_{k_{n+1}}$. We then update the adversarial example based on $x_{i,K}$ along the direction of maximizing the loss function in meta-test step:

$$x_{i,mt} = x_{i,K} + \beta \cdot \text{sign}(\nabla_{x_{i,K}} \mathcal{L}_{M_{k_{n+1}}}(x_{i,K})), \quad (16)$$

where β is the step size in meta-test step.

To improve the transferability of the adversarial example, as shown in Fig. 1, we add the perturbation obtained in the meta-test step to the adversarial example generated in the previous iteration x_i to update the adversarial example:

$$x_{i+1} = x_i + (x_{i,mt} - x_{i,K}). \quad (17)$$

The iteration of meta-train and meta-test can be conducted by a total of T times to generate the final adversarial example, where each iteration i randomly picks different models to establish various tasks and takes the output of the previous iteration $i-1$ as input. The procedure of MGAA is summarized in Algorithm 1.

It deserves to mention that the (n+1)-th model is not the black-box model in the real testing scenario. Actually, it is just a simulated black-box model to adaptively narrow the gaps of gradient directions between the meta-train and meta-test steps by iteratively simulating the white-box and black-box attacks.

Algorithm 1 Meta Gradient Adversarial Attack

Input:
the input example x_1
the classifier models $M_1, M_2, \cdots, M_N$
Output: the adversarial example x_{T+1}

for i $\in \{1, \cdots T\}$ **do**
  Randomly sample $n+1$ models M_{k_1}, M_{k_2}, $\cdots$, $M_{k_{n+1}}$ from $M_1, M_2, \cdots, M_N$ as a task
  $x_{i,0} = x_i$
  for j $\in \{0, 1, \cdots K-1\}$ **do**
    Calculate the cross entropy loss of input $x_{i,j}$ under the ensemble of n models $M_{k_1}, M_{k_2}, \cdots, M_{k_n}$: $\mathcal{L}_{M_{k_1},\cdots,M_{k_n}}(x_{i,j})$ as Equ. (13)
    $x_{i,j+1} = x_{i,j} + \alpha \cdot \text{sign}(\nabla_{x_{i,j}} \mathcal{L}_{M_{k_1},\cdots,M_{k_n}}(x_{i,j}))$
  end for
  Calculate the cross entropy loss of input $x_{i,K}$ under the model $M_{k_{n+1}}$: $\mathcal{L}_{M_{k_{n+1}}}(x_{i,K})$ as Equ. (15)
  $x_{i,mt} = x_{i,K} + \beta \cdot \text{sign}(\nabla_{x_{i,K}} \mathcal{L}_{M_{k_{n+1}}}(x_{i,K}))$
  $x_{i+1} = x_i + (x_{i,mt} - x_{i,K})$
end for

3.3. Analysis

In this section, we give a detailed analysis on the reasons why our proposed architecture is effective theoretically. To be specific, we consider the objective function in the meta-test step:

$$\arg\max_x J(f_{n+1}(x + \alpha \nabla_x J(\sum_{i=1}^{n} f_i(x)/n, y)), y), \quad (18)$$

where J denotes the cross entropy loss, f_i denotes the i-th sampled model in the task, n denotes the number of models used in the meta-train step, f_{n+1} denotes the sampled model used in the meta-test step, x and y denote the adversarial example and the true class label of the benign example respectively. Similar to [23], with the first order Tayler expansion of Equ. (18) at point x, the objective function can be rewritten as:

$$\arg\max_x J(f_{n+1}(x), y) + \alpha \nabla_x J(\sum_{i=1}^{n} f_i(x)/n, y) \nabla_x J(f_{n+1}(x), y). \quad (19)$$

The first term in Equ. (19) can be regarded as the optimization of the meta-test step, and the second term can be regarded as the calculation of cosine similarity of two gradients, that is, constraining the gradient directions in meta-train and meta-test steps as similar as possible, which is consistent with our motivation of narrowing the gaps of the gradient directions between them to improve the transferability of generated adversarial examples. We also conduct experiments to verify that the directions of the generated adversarial perturbations by our proposed MGAA are more

similar to the gradients of the black-box model than existing methods, which are provided in the supplementary material.

The existing gradient-based methods only focus on the update of the gradient during the white-box attack, and cannot obtain any information about the black-box attack. In contrast, our proposed architecture iteratively simulates the white-box attack and black-box attack, and constrains the gradient directions of the two to be similar in each iteration. In the iterative process, different combinations of models from the model zoo are sampled to compose different tasks, which help the adversarial examples access various model distributions when they are generated. The adversarial examples obtained after multiple iterations may not be task-specific optimal, *i.e.*, biased to any existing white-box models, but it generally contains better transferability for any unseen black-box model. This explains why our proposed architecture can improve the success rates of both white-box and black-box attacks.

3.4. Discussion

Differences from Meta-Attack [11]. In spite of the similarity in name, our proposed Meta Gradient Adversarial Attack (MGAA) is quite different from Meta-Attack [11]. Firstly, the tasks solved by these two methods are different. Meta-Attack solves the task of the score-based black-box attack, while our proposed MGAA addresses the task of the transfer-based black-box attack, which is more challenging. More importantly, Meta-Attack trains an extra gradient estimator as vanilla meta-learning methods do to estimate the gradient information of different input examples. Since the gradient estimator is a generative model, it can be quite difficult to learn. In contrast, no extra model is needed to train in our proposed MGAA architecture, which only borrows the idea of meta-learning and directly updates the adversarial examples through gradients in the iterative process. By easily integrating with the gradient-based methods, which are dominating the current black-box attack field, in a plug-and-play mode, our MGAA architecture has greater potential to generate more aggressive adversarial examples.

Differences from existing ensemble strategies. There have been several ensemble-based adversarial attack methods. We give a brief discussion on the differences between our MGAA and them. Liu *et al*. [30] proposes an optimization-based method to generate adversarial examples by the ensemble of models, which is totally different from our framework integrated with the gradient-based attack method. MIM [8] proposes an efficient ensemble strategy for gradient-based attack methods. By analyzing three kinds of ensemble strategies of the models, MIM finds that ensemble the results of models by the logits layer is the most effective way to generate adversarial examples. Different from the simple way to summarize the logits layer of all the models, we propose a more advanced architecture, which gradually narrows the gap of update directions between white-box attack and black-box attack settings to further improve the transferability of adversarial examples.

4. Experiment

In this section, we conduct extensive experiments to verify the effectiveness of the proposed Meta Gradient Adversarial Attack architecture. We first introduce the settings in the experiments, including the datasets, models, and experiment details in Sec. 4.1. Then some experiments are performed to investigate the impact of different hyperparameters in the Meta Gradient Adversarial Attack architecture in Sec. 4.2. Moreover, our proposed architecture is compared with the state-of-the-art methods, demonstrating the superiority of our Meta Gradient Adversarial Attack method under both the targeted and untargeted setting in Sec. 4.3. The ablation study of our MGAA is provided in the supplementary material, which demonstrates the effects of the meta-train and the meta-test step, respectively. Moreover, the experiments of attack under various perturbation budgets and the mininum adversarial noises used to attack are provided in the supplementary material.

4.1. Settings

Datasets. We use ImageNet [38] and CIFAR10 [20] datasets to conduct the experiments. For ImageNet, the ImageNet-compatible dataset[1] [38] in the NIPS 2017 adversarial competition is used, which contains 1000 images with a resolution of $299 \times 299 \times 3$. A baseline model of Inceptionv3 [44] can achieve a classification accuracy of 100% on these images. For CIFAR10, the test set with 10000 images is evaluated in our experiments.

Models. Our architecture utilizes a total of 10 white-box models to generate adversarial examples. In each iteration, multiple models are randomly selected to compose a meta-task. Under a black-box attack scenario, we evaluate 6 and 7 models on ImageNet and CIFAR10, respectively. All the models used in white-box and black-box settings are shown in Tab. 1. The details of these models are provided in the supplementary material.

Experiment Details. The range of pixel value in each image is 0-255, and our maximum perturbation ϵ is set to 16 on ImageNet and 8 on CIFAR10. We compare our proposed architecture with MIM [8], DIM [50], TIM [9] and SI-NI [27] methods. For MIM, we adopt the decay factor $\mu = 1$. For DIM, the transformation probability is set to 0.7. For TIM, the size of the Gaussian kernel is set to 7×7. For SI-NI, the number of scale copies is set to $m = 5$. For all iterative methods, including ours, the number of iteration T

[1] https://github.com/tensorflow/cleverhans/tree/master/examples/nips17_adversarial_competition/dataset

Table 1: Models on CIFAR10 and ImageNet. The first 10 models are treated as white-box models and the rest models are black-box models in our experimental settings.

	No.	ImageNet	CIFAR10
white-box models	1	Inceptionv3 [44]	ResNet-18 [17]
	2	Inceptionv4 [42]	ResNetv2-18 [18]
	3	InceptionResNetv2 [42]	GoogLeNet [43]
	4	ResNetv2-152 [18]	ResNeXt-29 [51]
	5	Ens3_Inceptionv3 [45]	SENet-18 [19]
	6	Ens4_Inceptionv3 [45]	RegNetX-200mf [36]
	7	Ens_InceptionResNetv2 [45]	DLA [52]
	8	ResNetv2-101 [18]	Shake-ResNet-26_2x64d [14]
	9	MobileNetv2_1.0 [40]	Adv_ResNet-18
	10	PNasNet [28]	Adv_DenseNet-121
black-box models	11	Adv_Inceptionv3 [45]	PyramidNet-164 [16]
	12	NasNet_mobile [53]	CbamResNeXt [47]
	13	MobileNetv2_1.4 [40]	Adv_GoogLeNet
	14	R&P [49]	Adv_ResNet-18_ll
	15	NIPS-r3	k-WTA [48]
	16	CERTIFY [6]	GBZ [24]
	17		ADP [35]

is set to 40. Unless mentioned, all experiments in this section are based on the integration of TI-DIM [9] method with our proposed architecture. The step sizes in meta-train step α and meta-test step β are 1 and ϵ/T, respectively. All the experiments of baseline methods [50, 9, 27] in our paper all adopt the ensemble strategy in MIM [8] since it is the most effective strategy available.

4.2. Impact of Hyperparameters

In our proposed architecture of Meta Gradient Adversarial Attack (MGAA), three hyperparameters may affect attack success rates: the number of models n selected for the ensemble-based attacks during the meta-train step, the number of iterations K during the meta-train step, and the number of tasks T sampled during the entire generation. We analyze in detail how does each hyperparameter affect the final performance of our proposed architecture. Due to the space limitations, we only elaborate the analysis of K here. The results and detailed analysis of n and T are provided in the supplementary material.

The number of iterations K in the meta-train step. In the meta-train step, the number of iteration steps K used to iteratively updating the adversarial example plays an important role in improving the success rates by the ensemble of multiple models. We compare the attack success rates of generated adversarial examples against white-box and black-box models with different iteration steps K in Tab. 2. With more iteration steps, the generated adversarial examples are more aggressive. When these aggressive adversarial examples are used as the basis for the meta-test step, the transferability of perturbation obtained by the meta-test step can also be stronger. But at the same time, the more iteration steps mean the more time it takes to generate adversarial examples. We recommend the value of K to be 5 in the following experiments.

The number of sampled tasks T. The more sampled tasks are taken, the higher attack success rate can be achieved, especially for black-box settings. However, on the other hand, increasing the number of sampled tasks also increases the time it takes to generate adversarial examples. Considering a trade-off of both efficiency and effectiveness, the value of T is recommended to be 40. The detailed results are provided in the supplementary material.

The number of ensembled models n in meta-train step. When the number of ensembled models increases, the success rates against the white-box and black-box attacks become higher and higher. But when n is greater than 5, the increase in attack success rates is not obvious. Considering that the more ensembled models in each iteration, the higher the computational complexity is needed, the number of ensembled models being 5 is a suitable choice. The detailed results are provided in the supplementary material.

4.3. Compared with the State-of-the-art Methods

4.3.1 The Targeted Attack

The experiments of adversarial examples generated on ImageNet under the targeted attack setting are shown in Tab. 3. The target label of each image is provided by the dataset. Compared to the baseline methods, the attack success rate significantly increases when integrating baseline methods with our proposed MGAA. For the DIM method [50], the integration with MGAA brings an average increase of 27.67% and 28.52% in white-box and black-box settings, respectively, reaching 95.10% and 47.90% respectively.

4.3.2 The Untargeted Attack

We compare our proposed MGAA with some typical gradient-based attack methods on ImageNet and CIFAR10 in Tab. 4 and Tab. 5 respectively under untargeted attack setting. Both results show the superiority of our proposed MGAA. It can be seen from Tab. 4 that although the SI-NI method [27] is effective in generating adversarial examples from a single model for the transfer-based black-box attack, the result is not satisfactory enough when using the ensemble of multiple models to conduct black-box attacks. Also, this method is time-consuming and memory-consuming due to the need for multiple scale copies of the model, which is unsuitable for the ensemble of multiple models. Thus we do not integrate it with our MGAA. Integrating other existing methods (*e.g.* MIM, DIM, and TI-DIM) with our proposed MGAA architecture, the attack success rates against the white-box and black-box models have been consistently improved.

Tab. 4 also gives the time cost of our MGAA on a GTX 1080Ti GPU. Although MGAA performs relatively slower when integrated with the existing gradient-based attack method, our MGAA achieves higher attack success rates when the T is 40. We also conduct the experiments

Table 2: The attack success rates of the adversarial examples generated under **different number of iterations** K in meta-train step against the white-box and black-box models on ImageNet. The number of sampled tasks T is 40. The number of ensembled models n in the meta-train step is 5. The index of models in the table is the same as Tab. 1.

K	white-box models										black-box models					
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	99.1	99.2	97.7	98.1	98.7	98.8	97.2	98.1	98.6	98.5	96.3	97.7	97.5	95.9	96.3	68.2
2	99.5	99.5	98.9	98.5	99.2	99.0	98.2	98.6	99.0	98.9	97.4	98.2	98.3	97.7	97.6	70.2
5	99.9	**100**	99.7	99.5	99.8	99.7	98.9	99.5	99.5	99.7	98.6	99.3	99.1	98.7	98.6	71.3
8	**100**	**100**	**99.9**	**99.8**	**99.9**	**99.8**	**99.4**	**99.8**	**99.6**	**100**	**99.1**	**99.4**	**99.5**	**99.4**	**99.3**	**71.6**

Table 3: The success rates under **targeted attack** setting on **ImageNet**. The number of ensembled models n in meta-train step is 5. The number of iterations K in meta-train step is 5.

n	white-box models											black-box models					
	1	2	3	4	5	6	7	8	9	10	avg.	11	12	13	14	15	avg.
MIM [8]	96.7	80.8	68.3	59.7	96.2	97.0	78.0	65.5	77.1	38.5	75.78	0.0	6.9	6.8	0.3	0.2	2.84
MGAA w/ MIM	**99.7**	**99.2**	**96.9**	**93.5**	**99.2**	**99.7**	**96.9**	**94.9**	**98.6**	**84.7**	**96.33**	0.0	**19.9**	**19.1**	**1.8**	**2.0**	**8.56**
DIM [50]	84.7	74.8	70.1	59.7	74.2	76.2	47.9	60.8	73.4	52.5	67.43	0.8	32.4	30.4	16.7	16.8	19.42
MGAA w/ DIM	**99.4**	**98.3**	**96.3**	**92.5**	**96.4**	**97.5**	**87.9**	**94.4**	**97.3**	**91.0**	**95.10**	**3.5**	**65.5**	**62.7**	**54.2**	**53.6**	**47.90**
TI-DIM [9]	54.8	46.2	39.0	47.2	39.5	40.1	32.1	20.3	49.6	40.4	42.85	**39.6**	23.7	14.5	34.2	26.6	23.86
MGAA w/ TI-DIM	**96.0**	**90.3**	**82.8**	**88.6**	**82.2**	**83.0**	**71.7**	**91.5**	**88.7**	**80.4**	**85.52**	37.3	**40.7**	**24.5**	**44.4**	**44.0**	**38.18**

Table 4: The success rates under **untargeted attack** setting on **ImageNet**. The number of ensembled models n in meta-train step is 5. The number of iterations K in meta-train step is 8.

Method	white-box models										black-box models						Time
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	(*s*/img)
SI-NI [27]	99.7	97.5	97.4	96.3	98.8	98.6	90.8	95.6	99.7	98.2	48.2	90.8	92.9	50.6	58.5	38.9	68.29
MIM [8]	99.6	99.7	99.4	98.7	99.8	99.8	99.5	99.0	99.1	98.2	44.4	92.6	94.1	65.4	72.2	34.4	17.51
MGAA w/ MIM	**100**	**100**	**100**	**99.9**	**100**	**100**	**100**	**99.9**	**99.9**	**99.9**	**52.0**	**96.0**	**96.9**	**67.1**	**74.9**	**37.0**	71.24
DIM [50]	99.4	99.8	99.5	98.6	99.4	99.5	98.5	98.6	98.9	98.8	79.4	98.0	98.3	95.0	95.3	44.8	22.22
MGAA w/ DIM	**100**	**100**	**100**	**99.9**	**100**	**100**	**99.9**	**99.9**	**100**	**99.9**	**88.0**	**99.9**	**99.8**	**98.9**	**98.9**	**49.3**	69.26
TI-DIM [9]	98.9	99.1	98.2	98.3	98.9	98.6	97.3	98.0	98.1	98.3	96.3	97.5	97.5	96.7	96.8	67.8	19.13
MGAA w/ TI-DIM	**100**	**100**	**99.9**	**99.8**	**99.9**	**99.8**	**99.4**	**99.8**	**99.6**	**100**	**99.1**	**99.4**	**99.5**	**99.4**	**99.0**	**71.6**	67.28

Table 5: The success rates under **untargeted attack** setting on **CIFAR10**. The number of ensembled models n in meta-train step is 5. The number of iterations K in meta-train step is 5.

Method	white-box models										black-box models						
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
MIM [8]	99.84	99.96	99.99	99.98	99.94	99.76	**99.89**	99.72	80.39	96.68	98.49	99.56	90.55	39.20	96.34	82.92	96.66
MGAA w/ MIM	**99.99**	**100**	**100**	**99.99**	**99.99**	**99.91**	99.87	**99.87**	**85.86**	**99.19**	**99.35**	**99.97**	**97.49**	**55.94**	**97.09**	**87.20**	**97.69**
DIM [50]	99.86	99.98	99.98	99.99	99.98	99.75	99.90	99.81	80.71	96.45	99.03	99.68	93.34	45.62	96.38	85.65	96.95
MGAA w/ DIM	**99.98**	**100**	**100**	**100**	**99.99**	**99.87**	**99.99**	**99.93**	**85.07**	**98.88**	**99.48**	**99.89**	**98.71**	**68.22**	**97.26**	**90.07**	**98.84**
TI-DIM [9]	99.86	99.96	99.98	99.99	99.97	99.76	99.94	99.85	81.17	96.66	99.27	99.74	93.61	46.69	96.50	85.76	96.92
MGAA w/ TI-DIM	**99.95**	**100**	**100**	**100**	**100**	**99.93**	**99.98**	**99.90**	**85.63**	**99.19**	**99.51**	**99.99**	**98.76**	**68.77**	**97.35**	**90.04**	**98.62**

with T being 10 in the supplementary material, and similar conclusions can be obtained. Further, when comparing our results with T being 10 and the results of TI-DIM with T being 40, we can see that the time consumed is nearly equal, but our method achieves higher attack success rates on both white-box and black-box settings.

5. Conclusion

Inspired by the philosophy of meta-learning, we propose a novel architecture named Meta Gradient Adversarial Attack to improve the transferability of adversarial examples. By iteratively simulating the scenarios of white-box and black-box attacks in the process of generating adversarial examples, the gap of gradient directions between black-box and white-box models can be reduced. Our architecture can be combined with any existing gradient-based attack method in a plug-and-play mode. Extensive experiments demonstrate that the adversarial examples generated by our architecture achieve better transferability. In future work, we will explore how to further improve the time efficiency of generating adversarial examples and analyze the reason why our method shows significant improvement under the targeted attack setting.

6. Acknowledgment

This work is partially supported by National Key R&D Program of China (No. 2017YFA0700800), Natural Science Foundation of China (Nos. 61806188, 61976219), and Shanghai Municipal Science and Technology Major Program (No. 2017SHZDZX01).

References

[1] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint arXiv:1712.04248*, 2017. 3

[2] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy (SP)*, pages 39–57, 2017. 1, 3

[3] 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 *ACM Workshop on Artificial Intelligence and Security (AISec)*, pages 15–26, 2017. 3

[4] Minhao Cheng, Thong Le, Pin-Yu Chen, Jinfeng Yi, Huan Zhang, and Cho-Jui Hsieh. Query-efficient hard-label black-box attack: An optimization-based approach. *arXiv preprint arXiv:1807.04457*, 2018. 3

[5] Shuyu Cheng, Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Improving black-box adversarial attacks with a transfer-based prior. In *Adv. Neural Inform. Process. Syst.*, pages 10934–10944, 2019. 3

[6] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. pages 1310–1320. PMLR, 2019. 7

[7] Guneet S Dhillon, Kamyar Azizzadenesheli, Zachary C Lipton, Jeremy Bernstein, Jean Kossaifi, Aran Khanna, and Anima Anandkumar. Stochastic activation pruning for robust adversarial defense. *arXiv preprint arXiv:1803.01442*, 2018. 3

[8] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 9185–9193, 2018. 1, 3, 4, 6, 7, 8

[9] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4312–4321, 2019. 1, 2, 3, 4, 6, 7, 8

[10] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7714–7722, 2019. 3

[11] Jiawei Du, Hu Zhang, Joey Tianyi Zhou, Yi Yang, and Jiashi Feng. Query-efficient meta attack to deep neural networks. *arXiv preprint arXiv:1906.02398*, 2019. 3, 6

[12] Gintare Karolina Dziugaite, Zoubin Ghahramani, and Daniel M Roy. A study of the effect of jpg compression on adversarial images. *arXiv preprint arXiv:1608.00853*, 2016. 3

[13] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*, 2017. 3

[14] Xavier Gastaldi. Shake-shake regularization. *arXiv preprint arXiv:1705.07485*, 2017. 7

[15] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 1, 2, 4

[16] Dongyoon Han, Jiwhan Kim, and Junmo Kim. Deep pyramidal residual networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5927–5935, 2017. 7

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 770–778, 2016. 7

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Eur. Conf. Comput. Vis.*, pages 630–645, 2016. 7

[19] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7132–7141, 2018. 7

[20] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 2, 6

[21] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016. 1, 2, 4, 5

[22] Alexey Kurakin, Ian Goodfellow, Samy Bengio, Yinpeng Dong, Fangzhou Liao, Ming Liang, Tianyu Pang, Jun Zhu, Xiaolin Hu, Cihang Xie, et al. Adversarial attacks and defences competition. In *The NIPS'17 Competition: Building Intelligent Systems*, pages 195–231. 2018. 3

[23] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Learning to generalize: Meta-learning for domain generalization. *arXiv preprint arXiv:1710.03463*, 2017. 5

[24] Yingzhen Li, John Bradshaw, and Yash Sharma. Are generative classifiers more robust to adversarial attacks? pages 3804–3814. PMLR, 2019. 7

[25] Yandong Li, Lijun Li, Liqiang Wang, Tong Zhang, and Boqing Gong. Nattack: Learning the distributions of adversarial examples for an improved black-box attack on deep neural networks. *arXiv preprint arXiv:1905.00441*, 2019. 3

[26] Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, and Jun Zhu. Defense against adversarial attacks using high-level representation guided denoiser. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1778–1787, 2018. 3

[27] Jiadong Lin, Chuanbiao Song, Kun He, Liwei Wang, and John E Hopcroft. Nesterov accelerated gradient and scale invariance for adversarial attacks. In *Int. Conf. Learn. Represent.*, 2020. 1, 3, 4, 6, 7, 8

[28] Chenxi Liu, Barret Zoph, Maxim Neumann, Jonathon Shlens, Wei Hua, Li-Jia Li, Li Fei-Fei, Alan Yuille, Jonathan Huang, and Kevin Murphy. Progressive neural architecture search. In *Eur. Conf. Comput. Vis.*, pages 19–34, 2018. 7

[29] Xuanqing Liu, Minhao Cheng, Huan Zhang, and Cho-Jui Hsieh. Towards robust neural networks via random self-ensemble. In *Eur. Conf. Comput. Vis.*, pages 369–385, 2018. 3

[30] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. *arXiv preprint arXiv:1611.02770*, 2016. 6

[31] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 1, 2, 3

[32] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2574–2582, 2016. 1, 2

[33] Y. Nesterov. A method for unconstrained convex minimization problem with the rate of convergence $o(1/k^2)$. 1983. 4

[34] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018. 3

[35] Tianyu Pang, Kun Xu, Chao Du, Ning Chen, and Jun Zhu. Improving adversarial robustness via promoting ensemble diversity. pages 4970–4979. PMLR, 2019. 7

[36] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10428–10436, 2020. 7

[37] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Certified defenses against adversarial examples. *arXiv preprint arXiv:1801.09344*, 2018. 3

[38] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.*, 115(3):211–252, 2015. 2, 6

[39] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *arXiv preprint arXiv:1805.06605*, 2018. 3

[40] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4510–4520, 2018. 7

[41] Chuanbiao Song, Kun He, Jiadong Lin, Liwei Wang, and John E Hopcroft. Robust local features for improving the generalization of adversarial training. *arXiv preprint arXiv:1909.10147*, 2019. 3

[42] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. *arXiv preprint arXiv:1602.07261*, 2016. 7

[43] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1–9, 2015. 7

[44] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2818–2826, 2016. 6, 7

[45] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017. 3, 7

[46] Eric Wong, Frank Schmidt, Jan Hendrik Metzen, and J Zico Kolter. Scaling provable adversarial defenses. In *Adv. Neural Inform. Process. Syst.*, pages 8400–8409, 2018. 3

[47] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Eur. Conf. Comput. Vis.*, pages 3–19, 2018. 7

[48] Chang Xiao, Peilin Zhong, and Changxi Zheng. Enhancing adversarial defense by k-winners-take-all. *arXiv preprint arXiv:1905.10510*, 2019. 7

[49] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. *arXiv preprint arXiv:1711.01991*, 2017. 3, 7

[50] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2730–2739, 2019. 1, 3, 4, 6, 7, 8

[51] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1492–1500, 2017. 7

[52] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2403–2412, 2018. 7

[53] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 8697–8710, 2018. 7

[54] 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 *Eur. Conf. Comput. Vis.*, 2020. 3