Towards Multiple Black-boxes Attack via Adversarial Example Generation Network

Mingxing Duan[1,3], Kenli Li[1*], Lingxi Xie[2], Qi Tian[2], and Bin Xiao[3*]
[1]School of Information Science and Engineering, Hunan University
[2]Huawei Inc
[3]Department of Computing, Hong Kong Polytechnic University
Email: {duanmingxing, lkl*}@hnu.edu.cn, 198808xc@gmail.com, tian.qi1@huawei.com
b.xiao@polyu.edu.hk*

ABSTRACT

The current research on adversarial attacks aims at a single model while the research on attacking multiple models simultaneously is still challenging. In this paper, we propose a novel black-box attack method, referred to as MBbA, which can attack multiple black-boxes at the same time. By encoding input image and its target category into an associated space, each decoder seeks the appropriate attack areas from the image through the designed loss functions, and then generates effective adversarial examples. This process realizes end-to-end adversarial example generation without involving substitute models for the black-box scenario. On the other hand, adopting the adversarial examples generated by MBbA for adversarial training, the robustness of the attacked models are greatly improved. More importantly, those adversarial examples can achieve satisfactory attack performance, even if these black-box models are trained with the adversarial examples generated by other black-box attack methods, which show good transferability. Finally, extensive experiments show that compared with other state-of-the-art methods: (1) MBbA takes the least time to obtain the most effective attack effects in multi-black-box attack scenario. Furthermore, MBbA achieves the highest attack success rates in a single black-box attack scenario; (2) the adversarial examples generated by MBbA can effectively improve the robustness of the attacked models and exhibit good transferability.

CCS CONCEPTS

• **Security and privacy** → **Systems security**; *Security of deep learning models.*

KEYWORDS

Black-box Attacks; Multiple Models; Adversarial Examples; DNN

ACM Reference Format:
Mingxing Duan[1,3], Kenli Li[1*], Lingxi Xie[2], Qi Tian[2], and Bin Xiao[3*], [1]School of Information Science and Engineering, Hunan University, [2]Huawei Inc, [3]Department of Computing, Hong Kong Polytechnic University, Email: {duanmingxing, lkl*}@hnu.edu.cn, 198808xc@gmail.com, tian.qi1@huawei.com b.xiao@polyu.edu.hk. 2021. Towards Multiple Black-boxes Attack via Adversarial Example Generation Network. In *MM '21: ACM International Conference on Multimedia, Oct. 20-24, 2021.* ACM, New York, NY, USA, 9 pages.

1 INTRODUCTION

Deep neural networks (DNNs) are widely used in face recognition [33], target tracking [8], natural language processing [18] and other applications [21], [23]. However, many recent studies have shown that DNNs are vulnerable to adversarial examples. The so-called adversarial examples add imperceptible perturbations to benign examples to generate new samples, which in turn make DNNs deviate from the correct prediction results. Szegedy *et al.* [35] first proposed L-BFGS to generate adversarial examples, and then a growing number of scholars have conducted lots of in-depth research on adversarial attacks. After that, a series of adversarial attack methods are proposed, such as JSMA [28], Deepfool [26], etc. At present, adversarial attacks are mainly divided into two categories: one is white-box attack and the other is black-box attack. The research on white-box attack is relatively mature due to that everything about the model is known while for the black-box model, the attacker has no idea about its structures, training parameters, defense methods, etc. More importantly, real-world applications involve more black-box models, which makes this research still challenging and attractive.

In recent years, common black-box attacks are mainly divided into three main categories: transfer-based attack, scored-based attack, and decision-based attack. The transfer-based attack method does not attack the black-box model directly, but constructs a substitute model with a distribution close to the black-box model. After that, the white-box attack algorithms are used to attack the substitute model, such as AdvGAN [38] and DaST [39]. However, transfer-based method cannot ensure that the substitute model can learn the generalization and robust performance of the black-box model completely, resulting in a lower attack success rate [6]. The scored-based attack method calculates the prediction score and directly generates adversarial examples by estimating the gradient of the attacked model [6], [19]. The decision-based attack method adopts the idea of optimization with a predicted gradient to carry out attacks [3]. However, these methods all aim at attacking a single black-box system. The research on attacking multiple black-boxes

[*] Corresponding Author

MM '21, Oct. 20-24, 2021, Chengdu, China

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00

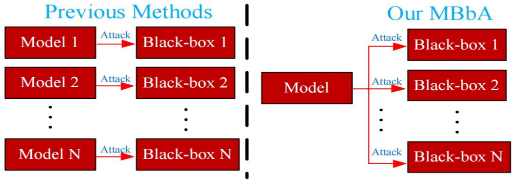


Figure 1: The difference between our MBbA and previous methods. Our unique model can attack multiple black-boxes simultaneously while previous methods need to train a distinct attack system to attack each black-box model.

is still full of challenges. For example, some bank access control systems require multiple identifications to pass before allowing the door to be opened, such as fingerprint, face, voice, and other biological characteristics of the same person [29]. In this case, the multi-model attack needs to be completed at one time to open the door.

A close problem to our multi-black-box attack is the multi-target attack in [15] which is the first to propose a multi-target attack algorithm MAN. MAN can generate adversarial examples to attack multiple categories directly and exhibits satisfactory performance, but this method cannot attack multiple systems at the same time. For multiple black-boxes, the network parameters of each model are unknown. Furthermore, input and output tasks of each model are different, so it is difficult to effectively design a system to attack multiple totally different models once. Therefore, our proposed method mainly attacks two types of multiple black-boxes. The first type has the same training dataset input, referred to as SI and the second type has the same output distribution, referred to as SO.

In this paper, we propose a novel black-box attack method called MBbA, which can generate multiple adversarial examples at one time. As shown in Fig. 1, MBbA can attack multi-black-box model simultaneously without training multiple models, which greatly reduces training costs. Compared with existing black-box attack algorithms, our proposed MBbA is practical. As illustrated in Fig. 2, MBbA first adopts multiple encoders to encode the input samples and target categories into associated spaces. Then these intermediate features are decoded into corresponding adversarial examples with the designed loss functions. This process is end-to-end generation for multi-black-box model instead of generating by substitute models. Extensive experiments show that according to input target categories, MBbA can efficiently seek the appropriate attack areas from the input image, and then perform an effective attack. This is also one of the main reasons that most of its attacks in each model are more effective than those in single-model attack methods. In addition, in terms of improving the robustness of the attacked models, the adversarial examples generated by MBbA are more effective than those generated by other state-of-the-art algorithms. MBbA shows good transferability. Our contributions are mainly as follows:

- To our best knowledge, this is the first work to study attacks on multi-black-box system and our MBbA adopts an end-to-end model to attack multiple systems once which takes less time to achieve the most effective attack model.
- Compared to state-of-the-art methods: 1) in single black-box attack scenario, MBbA obtains the highest attack success rates; 2) in multi-black-box attack scenario, our method takes the least time to achieve competitive and effective attack performance; 3) its success rates on attacking multiple models simultaneously are the best.
- The adversarial examples generated by MBbA not only demonstrate good transferability, but also effectively improve the robustness of multi-black-box model.

2 RELATED WORK

Because the attacker knows all the parameters of the white-box models, the researches on the white-box attacks are relatively mature, such as L-BFGS [35], FGSM [13], JSMA [28], DeepFool [26], C&W [4], etc. However, most of the realistic application systems are black-box models, and the white-box attack methods cannot be directly applied to these systems, so the black-box attack researches have attracted the interest of many scholars. At present, the black-box attack methods are mainly divided into three categories: transfer-based black-box attacks, decision-based black-box attacks, and score-based black-box attacks.

Transfer-based black-box attacks: The transfer-based attack algorithms make full use of the good transferability of adversarial examples. Papernot *et al.* [28], [13] constructed a model to replace the attacked model, and used the substituted model to generate adversarial examples to attack the black-box (attacked) model. Liu *et al.* [24] proved that the adversarial examples generated by the ensemble method have good transferability. Dong *et al.* [11] designed a momentum-based iterative algorithm to increase the success rate of the black-box attacks. Wang *et al.* [37] proposed a multi-stage network system for black-box attack by exploiting the features of different levels, which fully demonstrates that transferability plays an important role in the black-box attacks. However, the transfer-based attack methods have poor performance in target attacks [39], and in the real world, it is difficult to find a suitable substitute model to replace the black-box model.

Decision-based black-box attacks: Decision-based attack method was first proposed by Brendel *et al.* [3], which first constructs a large perturbation, and then slowly reduce the perturbation while maintaining the adversarial properties. Cheng *et al.* [19] regarded the black-box attack problem as a real-valued optimization problem and achieved good performance. Based on the outputs of the attacked model, Chen *et al.* [5] updated the direction of the gradient on the gradient boundary to generate the corresponding adversarial sample. Dong *et al.* [12] evaluated the robustness of face recognition systems adopting decision-based black-box attack method. The algorithms mentioned above have poor performance on ℓ_∞. To solve this problem, Chen *et al.* [7] proposed an efficient decision-based ℓ_∞ attack algorithm to improve the attack success rate via flipping the signs of a few entries in perturbations. At present, all works on decision-based attack algorithm are still limited to a single model, but it has not been applied to multi-model attack. The main reason is that in a multi-model system, each submodel requires different gradient learning, which results in great computational overhead, and it is difficult to ensure that every model converges.

Score-based black-box attacks: The score-based methods mainly continuously optimize the perturbed samples through the corresponding outputs or losses from querying the black-box models, so as to obtain suitable adversarial examples to achieve successful black-box attacks. Chen *et al.* [6] put forward a zeroth order optimization approach to generate adversarial examples. By querying the attacked models, Bhagoji *et al.* [2] proposed a novel Gradient Estimation black-box attack method to generate adversarial examples. An adaptive random gradient estimation method was proposed by Tu *et al.* [36] to balance query counts and distortion. Recently, many studies [14], [22], [34] have begun to speed up the black-box query process and achieved good results. Although these methods mentioned above show good performance in single black-box attack scenarios, there are still no studies on multi-black-box attack. Our MBbA is the first to propose the research on attacking multi-black-box model simultaneously.

3 METHODOLOGY

In this section, we first introduce the attack scenarios involved in this paper, then describe our MBbA model in detail, and finally present the optimization process of the entire system.

3.1 Attack Scenario

3.1.1 Non-target Attack vs. Target Attack . A non-target attack is to make the target model misclassify the perturbed sample while it does not specify which category it is classified into. However, in the target attack scenario, the generated sample is misjudged as a specified label by the attacked model. Due to that the non-target attack scenarios are relatively simple, our MBbA carries out the following study based on the target attacks.

3.1.2 Multi-black-box Attack Scenario. Most practical applications are basically black-box models, many of which seem to be unrelated. In fact, some black-box models are related to each other. For example, in the defense process of the COVID-19 [10] epidemic, the video surveillance system and the face swiping access control system can be applied to the same patient. Different systems of this type can be regarded as having the same input dataset, and the outputs may be same, partly same or completely different. The other type is that the inputs are not required to be same, but the outputs are same. For example, the face temperature recognition instrument repeatedly detects that the temperature of a person is so high to predict the person infected with COVID-19 while the intelligent medical system detects that the same person is infected with COVID-19 with nucleic acid analysis. These two types are mainly focuses of this paper. To facilitate identification, the scenario with the same input is referred to as the SI scenario and the scenario with the same output is referred to as the SO scenario.

3.2 Code for Target Label

To generate target adversarial examples faster and more effectively, we use the target label as an input to make the entire adversarial learning move in the direction of the target. Assuming that the real labels of the datasets in this paper are all represented as one-hot vectors, and given an image, its target label is also an one-hot vector. To facilitate training, we code the target label from a one-hot vector $\mathbf{z}$ to a three-dimensional tensor $\mathcal{Z} \in \mathcal{R}^{K\times H\times W}$, where H and W represent the height and width of a benign image, and K denotes the total label categories. In the same way as the one-hot vector, only one feature map is filled with ones, and the other feature maps are filled with zeros. Fig. 2 shows an example for that process.

3.3 Multi-black-box Attack

MBbA mainly attacks two types of multiple black-boxes which have the same input (SI) and the same output (SO). As indicated in Fig. 2, MBbA first transforms benign samples into adversarial examples through encoding and decoding processes based on the target categories. Then by querying the black-box models, they estimate whether the adversarial examples are misjudged as the specified targets. In addition, different loss functions are used to ensure that realistic and effective adversarial examples are generated, thereby increasing the attack success rates.

3.3.1 Adversarial Example Generation. As shown in Fig. 2, MBbA has two different function encoders, one is used to extract the features of the input images, and the other is to encode the target categories into the space associated with the input samples. After that, the encoded features are decoded to generate the corresponding adversarial examples through the optimization process.

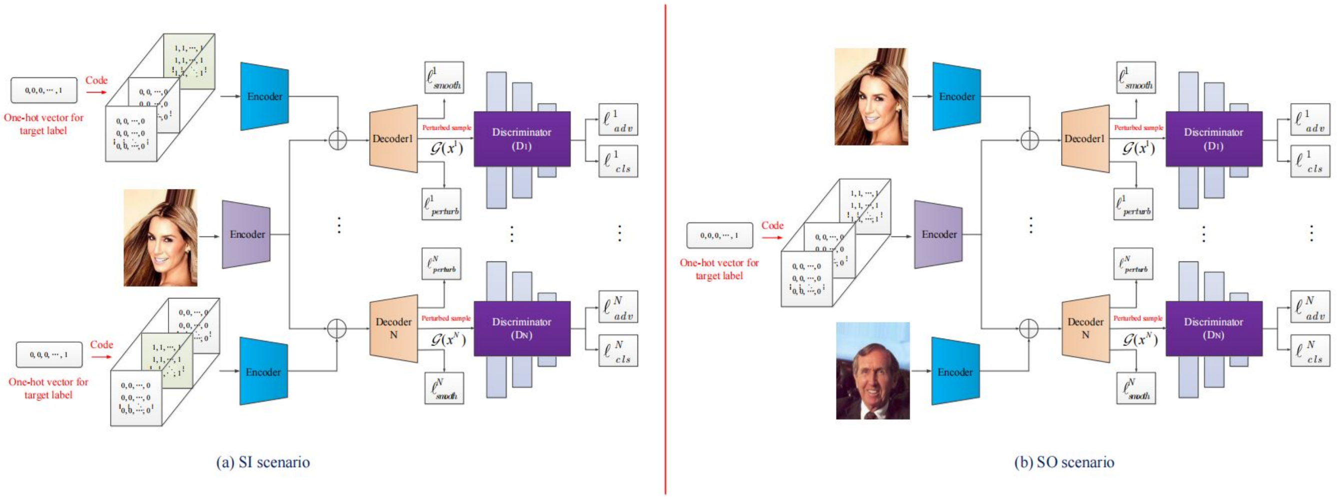
SI Scenario: Given an input sample $(\mathbf{x}_{SI}(i), \mathbf{y}_{SI}(i))$, i= {1, ..., M}, M represents the total number of samples, and the input category is a misjudgment label which is same or completely different. Of course, these are only two cases considered in this paper. When the adversarial sample is misjudged as the same target by all black-box models, the target label at this time is $\mathbf{y}'_{SI}(i)$ and the system is denoted as $\text{MBbA}^{S}_{\text{SI}}$. If the generated sample is misjudged as the different ones, the target labels at this time are $\{\mathbf{y}'^{1}_{SI}(i), ..., \mathbf{y}'^{N}_{SI}(i)\}$, where N represents the number of all categories, and this scenario is referred to as $\text{MBbA}^{D}_{\text{SI}}$. Therefore, the input samples and target labels are first transformed into

$$\begin{cases} \{E(\mathbf{x}_{SI}(i)),\ \underbrace{\{E_1(\mathbf{y}'_{SI}(i)),\ ...,\ E_N(\mathbf{y}'_{SI}(i))\}}_{\text{Same Target}}\ \} \\ \{E(\mathbf{x}_{SI}(i)),\ \underbrace{\{E_1(\mathbf{y}'^{1}_{SI}(i)),\ ...,\ E_N(\mathbf{y}'^{N}_{SI}(i))\}}_{\text{Different Targets}}\} \end{cases}, \qquad (1)$$

where $E(\cdot)$ and $\{E_1(\cdot), ..., E_N(\cdot)\}$ denote the corresponding encoders.

Then, we connect the encoded features according to the channel direction. For example, the intermediate features of a sample $E(\mathbf{x}_{SI}(i)) \in R^{C\times H\times W}$ and those of a target label $E_1(\mathbf{y}'^{j}_{SI}(i)) \in \mathcal{R}^{K\times H\times W}$ are merged to become $\{E(\mathbf{x}_{SI}(i)) + E_1(\mathbf{y}'^{j}_{SI}(i))\}$ $\in \mathcal{R}^{(K+C)\times H\times W}$. After that, an additional convolutional layer is used to adjust these fused features to obtain the final intermediate features with the size of $C \times H \times W$. Finally, these features are decoded into the corresponding adversarial examples.

SO Scenario: At this time, all black-box models have the same output distribution, and each misclassification target is same. Besides, only one encoder is used to encode the target label $\mathbf{y}'_{SO}(i)$, i = {1, ..., N}, where N represents the number of all categories. In this case, the input datasets can be divided into two categories in this paper: exactly same and completely different, so the intermediate



(a) SI scenario

(b) SO scenario

Figure 2: Full schematic diagram of MBbA. 'SI scenario' denotes that multiple black-boxes have the same training dataset input while 'SO scenario' means that multiple black-boxes have the same output distribution.

features are

$$\begin{cases} \{E(\mathbf{y}'_{SO}(i)),\ \underbrace{\{E_1(\mathbf{x}'_{SO}(i)),\ ...,\ E_Q(\mathbf{x}'_{SO}(i))\}}_{\text{Same Dataset}}\ \} \\ \{E(\mathbf{y}'_{SO}(i)),\ \underbrace{\{E_1(\mathbf{x}'^{1}_{SO}(i)),\ ...,\ E_Q(\mathbf{x}'^{Q}_{SO}(i))\}}_{\text{Different Datasets}}\} \end{cases}, \qquad (2)$$

where Q indicates the total number of different datasets, $\mathbf{x}'^{j}_{SO}(i)$ represents the ith sample in the jth dataset, and $\mathbf{x}'_{SO}(i)$ signifies the ith sample in the dataset. We need to emphasize that: 1) in the case of the same input dataset, all input images are the same each time and this scenario is denoted as $\text{MBbA}^{S}_{\text{SO}}$; 2) in the case of different datasets, to ensure a balanced learning process, we process the size of each dataset to ensure all datasets with the same size and this scenario is referred to as $\text{MBbA}^{D}_{\text{SO}}$. The decoding process is the same as those mentioned in SI scenario. It should be noted here that the two scenarios $\text{MBbA}^{S}_{\text{SI}}$ and $\text{MBbA}^{S}_{\text{SO}}$ have the same function, so we use MBbA(S) to express these two scenarios.

Smooth Loss: To reduce the influence of perturbations and make the generated adversarial examples look smoother [27], we use l_2 loss to alleviate the adversarial effect:

$$\ell^{j}_{smooth} = ||D_j(E(\mathbf{x}_m(i)), E(\mathbf{y}'_m(i))) - \mathbf{x}_m(i)||_2, \qquad (3)$$

where ℓ^{j}_{smooth} expresses the smooth loss of the jth decoder, D_j denotes the jth decoder, m indicates a certain scenario of SI or SO, and $\mathbf{x}_m(i)$ and $\mathbf{y}'_m$ respectively represent the ith sample in the mth scenario and its corresponding misclassified label. ℓ_{smooth} can also be used to ensure that the generated images keep the key information of benign images.

Perturbation Loss: To get good results with less perturbation, we adopt the method successfully applied in [24] [4], [38] to bound the magnitude of the perturbation, which is

$$\begin{cases} \ell^{j}_{perturb} = \mathbb{E}[\max(0,\ \Theta - c)] \\ \Theta = ||D_j(E(\mathbf{x}_m(i)), E(\mathbf{y}'_m(i))) - \mathbf{x}_m(i)||_2 \end{cases}, \qquad (4)$$

where $\ell^{j}_{perturb}$ signifies the perturbation loss of the jth decoder and c is a user-specified bound.

3.3.2 *Black-box Models as Discriminators and Classifiers* .

The black-box model has two main functions in this paper: the first one is as a discriminator, and the other is as a classifier.

Discriminator: When the black-box model is used as the discriminator (D), its main purpose is to judge the true and false of the generated images. Each generated adversarial sample must be queried through the corresponding black-box model to determine whether it meet the set requirements. Since D does not need to be trained and updated, the two-player game process of the traditional GAN only needs to consider the generator (G) learning (we need to emphasize that each generator consists of a decoder and corresponding two encoders), so the loss is

$$\ell^{j}_{GAN-G} = \mathbb{E}[\log(1 - Dis_j(D_j(E(\mathbf{x}_m(i),\ E(\mathbf{y}'_m(i))))))], \qquad (5)$$

where ℓ^{j}_{GAN-G} and Dis_j represent the losses of the jth generator and discriminator, respectively.

Classifier: The black-box models also serve as classifiers (C), which predict the classes of the generated samples. In this paper, we mainly focus on the problem of target attacks, so the black-box model has to determine whether the generated adversarial examples are misjudged into the specified categories. At this point, the classification loss can be expressed as

$$\begin{cases} \ell_{cls}^{j} = -\frac{1}{I}\sum_{i=1}^{I}\begin{pmatrix} \mathbf{y}'_m(i) In\ C_j(\varphi) + \\ (1-\mathbf{y}'_m(i)) In\ (1-C_j(\varphi)) \end{pmatrix}, \\ \varphi = D_j(E(\mathbf{x}_m(i),\ E(\mathbf{y}'_m(i)))) \end{cases} \tag{6}$$

where ℓ_{cls}^{j} represents the classification loss of the jth black-box model, C_j denotes the jth black-box model, and I is the batch size.

3.4 Optimization

The parameter update process of the entire system is mainly divided into two parts, the first part is the parameter learning of the sub-encoder-decoder, such as (E_1, D_1), ..., (E_N, D_N), the other part is to update the parameters of a separate encoder E. The optimization goal of the first part is

$$\ell_j = \ell_{cls}^{j} + \alpha\ell_{GAN-G}^{j} + \beta\ell_{perturb}^{j} + \gamma\ell_{smooth}^{j}, \tag{7}$$

where ℓ_j denotes the optimization function of the jth sub-coder-decoder, α, β, and γ are used to control the importance of each loss function, ℓ_{cls}^{j} is used to generate the target sample, ℓ_{GAN-G}^{j} makes the generated sample conform to the distribution of the input dataset, $\ell_{perturb}^{j}$ limits the magnitude of the perturbation, and ℓ_{smooth}^{j} weakens the impact of adversarial perturbation. Following the common practice in adversarial example generation [38], [15], [1], [39], α, β, and γ are set to 0.5, 0.88, and 0.6, respectively.

The parameter optimization function of the independent encoder E is

$$\nabla_E = \frac{1}{N}(\nabla_{\theta_1}\ell_1(\theta_1,\ D_1)+,\ ...,\ +\nabla_{\theta_N}\ell_N(\theta_N,\ D_N)), \tag{8}$$

where ∇_E represents the afferent gradient of E, which is used to update all the parameters of E, N denotes the total number of the black-box models, and $\nabla_{\theta_j}\ell_j(\theta_j,\ D_j)$ expresses the gradient passed to the input of the jth decoder.

4 EXPERIMENTS

4.1 Experimental Settings

In this part, we introduce in detail the experimental settings, such as datasets, implementation details, and target models.

Datasets: IMDB-WIKI [31] (523,051 images, label: age and gender), CelebA [25] (202,599 images, label: 40 binary attributes annotations), and Morph-II [30] (55,000 images, label: age, gender, and race) are used to verify the performance of our MBbA. In the single-target attack and **SI** scenarios, since the input datasets are same, all images in each dataset are used. In the **SO** scenario, to ensure that the entire training is balanced, we need to ensure that the sizes of different input datasets are nearly same. We first utilize the data augmentation approach proposed in Ref. [16] to augment the number of Morph-II, and then, more than 200,000 images are achieved. Finally, we randomly select 200,000 images from each dataset as the new input datasets.

Implementation Details: We utilize the similar structure of CycleGAN [40] as the encoder and decoder, and the size of the convolution kernel of the additional convolutional layer is 1 × 1, and its output is 256. c is calculated as the method proposed in MAN [15], that is $c = \delta\sqrt{\kappa}$, where $\kappa = C\times H\times W$ represents the dimension of the input image. We set $\delta = 12$ in the section of adversarial attack and will analyze it in detail in the section of ablation studies. Adam optimizer with $\beta_1 = 0.5$ and $\beta_2 = 0.999$ is applied to train MBbA and the batch size is 64. All experiments are performed on NVIDIA Tesla P100.

Target Models: VGG16 [32], VGG19 [32], and ResNet34 [17] are used as the attacked models in this paper and their training processes in different scenarios will be explained in the following corresponding sections.

4.2 Adversarial Attack

4.2.1 Single Black-box Attack. In this section, we verify the attack effect of our MBbA on a single black-box model and compare it with state-of-the-art methods AdvGAN [38], MAN [15], and AI-GAN [1]. In the two variants of MAN, we choose the MANc model because it performs the best attack performance on ImageNet [9]. The predicted label at this time is gender, and the input target category is opposite to the real label of the input image. Morph-II, CelebA, and IMDB-WIKI are divided randomly according to the ratio of 3 (training):1 (verification):1 (test). At this time, the attacked models are VGG16 and VGG19, both of which are pre-trained on the corresponding datasets, the test accuracies of VGG16 on Morph-II, CelebA, and IMDB-WIKI are 98.6%, 99.2%, and 97.9%, respectively, while those of VGG19 are 99.1%, 99.5%, and 99.3%, respectively. For AdvGAN and AI-GAN, we adopt dynamic distillation method to learn the substitute model, and to make the whole experiments more convincing, the substitute model and the attacked model adopt the same network structure. In addition, c is $12\sqrt{\kappa}$.

On Morph-II, CelebA, and IMDB-WIKI, all models are trained for 100k, 200k, and 300k iterations, respectively, and the initial learning rate is 0.002. On Morph-II, when the whole iterations reach 80k, the learning rate is decreased by 10% every 10k iterations while that is reduced by 10 times every 20k iterations after 100k iterations on CelebA. When the iterations are 180k, the learning rate on CelebA is reduced by 10% every 10k iterations. On IMDB-WIKI, the learning rate begins to change when the iterations reach 200k, and it is reduced by 10% every 20k iterations. That will change to be decreased by 10% every 10k iterations when the iterations reach 280k. After AdvGAN, MAN, AI-GAN and MBbA are all trained, we use the test dataset to verify their performance. The success attack rate is the number of attack success samples divided by the total number of adversarial examples, which is as the evaluation standard. Each experiment is performed 10 times, and the corresponding average value is taken as the final result.

Table 1: Attack success rates of different algorithms on a single black-box attack scenario (%).

Algorithms	Morph-II		CelebA		IMDB-WIKI	
	VGG16	VGG19	VGG16	VGG19	VGG16	VGG19
AdvGAN	85.2	85.4	87.9	86.2	81.4	80.1
MAN	90.8	89.3	90.7	89.5	83.3	81.9
AI-GAN	88.4	87.2	88.9	87.4	82.3	81.4
MBbA	96.1	93.4	95.6	92.8	90.3	86.8

We show the attack success rates in Table 1. Compared with other algorithms, AdvGAN is less effective in target attacks, but because the predicted category is gender (male or female), it is a binary classification problem which means that the results on target attacks and on non-target attacks are almost same. As a consequence, the predictions of AdvGAN is close to other compared algorithms. In addition, in black-box attacks, both AdvGAN and AI-GAN need to train a substitute model, and then the generated adversarial examples are used to attack the substitute model. Since the substitute model cannot completely replace the original attacked model, their performance is weaker than the other two algorithms. Our proposed MBbA achieves the best attack performance among all compared algorithms, mainly because MBbA directly attacks the target model instead of a substitute model, and multiple loss functions and input target conditions ensure realistic and effective adversarial examples to be generated.

4.2.2 Multiple Black-boxes Attack. In this situation, we mainly considers three scenarios: $\mathrm{MBbA}_{\mathrm{SI}}^{D}$, $\mathrm{MBbA}_{\mathrm{SO}}^{D}$, and $\mathrm{MBbA}(S)$. What we need to point out is that the six categories of Young, Male, Eyeglasses, Mustache, Gray_Hair, and Bags_Under_Eyes are all binary classification problems (1 or 0), so the results of each category on target attacks and non-target attacks are basically same. In different scenarios, each attacked model is pre-trained with the corresponding dataset and labels. Table 2 shows the detailed pre-training results in different scenarios. The serial numbers ((a) ,(b), ...) indicate the corresponding training dataset or labels used by each attacked model, and the others express that the attacked models use the same training dataset or labels in different scenarios. The other settings in $\mathrm{MBbA}_{\mathrm{SI}}^{D}$, $\mathrm{MBbA}_{\mathrm{SO}}^{D}$, and $\mathrm{MBbA}(S)$ are the same as those on the single model attack scenario with CelebA, CelebA, and Morph-II, respectively. Tables 3, 4, and 5 present the attack success rates of different algorithms under different scenarios.

Table 2: The attacked models and test accuraies, training datasets, and output labels in different scenarios.

Scenarios	Attacked Models (Test Accuracies)	Datasets	Labels
$\mathrm{MBbA}_{\mathrm{SI}}^{D}$	(a) VGG16(99.1%), (b) VGG16(98.5%) (c) VGG19(98.7%), (d) VGG19(98.4%) (e) ResNet34(97.3%), (f) ResNet(97.2%)	CelebA	(a) Young, (b) Male (c) Eyeglasses, (d) Mustache (e) Gray_Hair, (f) Bags_Under_Eyes)
$\mathrm{MBbA}_{\mathrm{SO}}^{D}$	(a) VGG16(99.5%) (b) VGG19(99.2%) (c) ResNet34(98.1%)	(a) Morph-II (b) CelebA (c) IMDB-WIKI	Gender

$\mathrm{MBbA}(S)$	VGG16(98.6%), VGG19(99.1%)
	ResNet34(99.4%)
		Morph-II	Gender

Table 3: Attack success rates of different algorithms on multiple black-boxes in $\mathrm{MBbA}_{\mathrm{SI}}^{D}$ scenario (%).

Algorithms	VGG16		VGG19		ResNet34		Training Time for All Models(hours)
	Young	Male	Eye.	Mus.	Gray.	Bags.	
AdvGAN	81.8	85.4	74.6	75.3	67.3	63.8	201.3
MAN	87.2	91.3	79.9	77.6	65.2	71.9	190.1
AI-GAN	85.9	86.5	78.2	79.4	70.3	69.2	250.5
MBbA	89.1	90.7	83.2	81.6	68.9	67.2	45.7

From Tables 3, 4, and 5, we can conclude that with the same size of the training datasets, as the number of neural network layers deepens, the attack success rates gradually decrease. The main reason for these are that when the attacked models have been pre-trained, the deeper the model, the stronger its robust performance. Regardless of the different attack scenarios, when the attacked models are VGG16 or VGG19, our MBbA obtains the best performance among all comparison algorithms. In addition, although its attack performance on ResNet34 is not the best, it is close to the best performance achieved by comparison algorithms. They are specifically: in the $\mathrm{MBbA}_{\mathrm{SI}}^{D}$ scenario, the gaps are 1.4% and 4.7%, and in the $\mathrm{MBbA}_{\mathrm{SO}}^{D}$ and $\mathrm{MBbA}(S)$ scenarios, the gaps are 3.8% and 1.8%, respectively. The main reasons why MBbA can obtain satisfactory results are as follows: 1) the benign image can quickly find the areas that needs to be disturbed during the encoding and decoding process through the input targets and optimization functions, which allows it to generate target adversarial examples faster and more efficiently; 2) the four loss functions ensure that MBbA adopts the smallest and most effective perturbance to generate realistic adversarial examples.

Table 4: Attack success rates of different algorithms on multiple black-boxes in $\mathrm{MBbA}_{\mathrm{SO}}^{D}$ scenario (%).

Algorithms	VGG16 on Morph-II	VGG19 on CelebA	ResNet34 on IMDB-WIKI	Training Time for All Models(hours)
AdvGAN	87.2	85.6	63.4	96.3
MAN	91.6	89.7	68.2	87.2
AI-GAN	89.3	86.8	74.9	135.9
MBbA	92.4	89.7	71.1	34.5

Another obvious advantage of our MBbA is that it takes the least time to obtain satisfactory performance while ensuring that the most settings are the same as these in other comparison algorithms. In the $\mathrm{MBbA}_{\mathrm{SI}}^{D}$ scenario, MBbA spends 45.7 hours training on six different black-box models and achieves good test performance. AdvGAN, MAN, and AI-GAN need to be trained on each black-box model, so they need more time to complete these processes. In the end, the total training time for AdvGAN, MAN, and AI-GAN is 4.4, 4.16, and 5.48 times longer than that of MBbA, respectively. In the $\mathrm{MBbA}_{\mathrm{SO}}^{D}$ scenario, such ratios are 2.79 times, 2.53 times, and 3.94 times, respectively. In the $\mathrm{MBbA}(S)$ scenario, the four attack algorithms spend less time on training than those in the first two scenarios. The main reason is that the dataset is Morph-II, and the total number of images is greatly reduced. Meanwhile, MBbA still takes the least time to train, which is 35%, 40.5%, and 29.6% of the total training time of AdvGAN, MAN, and AI-GAN, respectively.

Table 5: Attack success rates of different algorithms on multiple black-boxes in $\mathrm{MBbA}(S)$ scenario (%).

Algorithms	VGG16 on Morph-II	VGG19 on Morph-II	ResNet34 on Morph-II	Training Time for All Models(hours)
AdvGAN	85.2	85.4	68.3	18.1
MAN	90.8	89.3	74.9	15.8
AI-GAN	88.4	90.1	77.6	21.6
MBbA	93.2	90.1	75.8	6.4

Finally, each attack success rate of all previous experiments represents the result on each attacked model. Next, we count images of this kind that are misjudged as the specified targets by all attacked models at the same time, and then calculate the ratios between them and the total number of adversarial examples to achieve the final attack success rates. Table 6 shows the attack success rates of the four attack algorithms in different scenarios.

Table 6: The attack success rates in the scenario where the same image is misjudged by all black-box models in the test datasets (%).

Scenario	AdvGAN	MAN	AI-GAN	MBbA
$\text{MBbA}_{\text{SI}}^{D}$	40.7	54.6	50.7	60.2
MBbA(S)	53.8	59.8	60.2	70.3

It can be seen from Table 6 that the attack success rates in this case are significantly lower than all previous results. The main reason is that the same image misjudged by all black-box models as the specified targets is more difficult than that misjudged by a single black-box model. Our MBbA achieves the best attack performance among all comparison algorithms. The main reason is that MBbA encodes the input sample and multiple input targets into associated spaces. Then through the optimization process, it can quickly exploit appropriate perturbance areas to generate effective adversarial examples. That process improves the attack success rates of multiple black-boxes simultaneously.

4.3 Adversarial Training

Adversarial training is one of the most effective ways to improve the robustness of the attacked systems, which trains these systems by the adversarial examples with the groundtruth labels. This section demonstrates the following highlights through adversarial training:

(1) MBbA takes the least time to improve the robustness of multiple models simultaneously;

(2) In the case of the same training dataset, MBbA generates the most effective adversarial examples among all comparison algorithms which show good transferability.

Setups: All experiments are implemented in the $\text{MBbA}_{\text{SI}}^{D}$ scenario, and all the training methods and settings are the same as those in the $\text{MBbA}_{\text{SI}}^{D}$ scene in **Section** 4.2.2. We randomly select 50,000 images from CelebA to generate six types of adversarial examples via pre-trained AdvGAN, MAN, AI-GAN, and MBbA. The six attacked models are fine-tuned on the corresponding adversarial examples, and their outputs are compared with the groundtruth labels. The maximum iterations is 80k, and the learning rate is 0.002 and it is decreased by 10% every 10k iterations when the iterations reach 60k. As the same approach used in Ref. [15], we verify the robustness and transferability brought by MBbA from the attack success rates.

Table 7: Attack success rates of different algorithms when the attacked models are fine-tuned on the adversarial examples generated by MBbA (%).

Attack Strength	Attack Methods	Fine-tuned on Adv. Exam. Generated by MBbA					
		Young	Male	Eye.	Mus.	Gray.	Bags.
$12\sqrt{\kappa}$	AdvGAN	12.9	13.1	10.8	9.7	8.3	8.1
	MAN	22.4	21.1	18.5	17.9	14.7	15.1
	AI-GAN	19.4	20.3	17.2	18.5	15.1	14.7
$16\sqrt{\kappa}$	AdvGAN	16.8	17.4	15.1	14.2	10.4	9.9
	MAN	27.2	24.9	21.1	21.5	13.9	16.9
	AI-GAN	24.1	23.8	20.9	22.3	16.7	17.1
$20\sqrt{\kappa}$	AdvGAN	19.7	20.5	18.7	16.9	12.1	12.3
	MAN	29.8	28.1	24.7	24.3	15.1	18.6
	AI-GAN	26.9	27.2	23.1	25.8	18.4	19.3

Table 8: Attack success rates of MBbA when the attacked models are fine-tuned on the adversarial examples generated by AdvGAN, MAN, and AI-GAN (%).

Attack Strength	Fine-tuned Methods	Attack Method with MBbA					
		Young	Male	Eye.	Mus.	Gray.	Bags.
$12\sqrt{\kappa}$	AdvGAN	25.2	26.4	22.8	21.9	21.2	20.7
	MAN	35.1	33.9	32.1	31	27.3	27.6
	AI-GAN	32.1	33.5	30.8	31.1	27.9	26.5
$16\sqrt{\kappa}$	AdvGAN	30.4	31.3	28.7	27.9	23.8	23.1
	MAN	41.2	39.5	34.9	34.9	28.2	30.1
	AI-GAN	38.4	37.1	34.2	35.8	29.2	30.2
$20\sqrt{\kappa}$	AdvGAN	34.1	34.9	33.2	30.7	26.4	27.8
	MAN	45.1	43.6	39.1	38.7	29.7	32.4
	AI-GAN	41.2	42.4	38.3	40.1	32.9	34.7

From Tables 7 and 8, we can find that with adversarial training, the overall defense performance of the attacked models is greatly improved, and the corresponding attack success rates drop sharply compared with the results achieved in the previous $\text{MBbA}_{\text{SI}}^{D}$ scenario. Furthermore, as the attack strength (c) increases, the attack success rates gradually increase. For example, when the attack strength is $12\sqrt{\kappa}$, the success rates of the adversarial examples misleading the attacked models are low, and when the attack strength increases to $20\sqrt{\kappa}$, the overall attack success rates increase.

In addition, we can draw the two important conclusions from Tables 7 and 8: (1) the adversarial examples generated by MBbA are more effective; (2) the adversarial examples generated by MBbA perform good transferability. In Table 7, when the attacked models are fine-tuned on the adversarial examples generated by MBbA, the attack success rates with the adversarial examples generated by AdvGAN, MAN, and AI-GAN do not exceed 30% even when the attack strength is $20\sqrt{\kappa}$. On the contrary, when the black-box models are fine-tuned on the adversarial examples generated by AdvGAN, MAN, and AI-GAN, the overall attack success rates with the adversarial examples generated by MBbA are improved greatly, which are 10% higher than those in Table 7. For example, the best attack success rate in Table 7 is 29.8%, while the corresponding result in Table 8 is 45.1%. Therefore, we can infer that under the same setting, the adversarial examples generated by MBbA are more effective than those generated by other attack algorithms, which can effectively improve the robustness of the attacked systems. More importantly, when the adversarial examples generated by MBbA in Table 8 or generated by AdvGAN, MAN, and AI-GAN in Table 7 are used to attack the fine-tuned models, the former achieves good attack success rates, which performs good transferability.

Table 9: The time for generating 300,000 adversarial examples with different attack methods.

Methods	AdvGAN	MAN	AI-GAN	MBbA
Time(hours)	17.4	16.9	17.9	2.7

Finally, Table 9 shows the time required for AdvGAN, MAN, AI-GAN, and MBbA to generate 300,000 adversarial examples. We can observe that MBbA only takes 2.7 hours to accomplish that task, while AdvGAN, MAN, and AI-GAN need 17.4 hours, 16.9 hours, and 17.9 hours, respectively. Therefore, we can conclude that our MBbA takes the least time to obtain the effective adversarial examples.

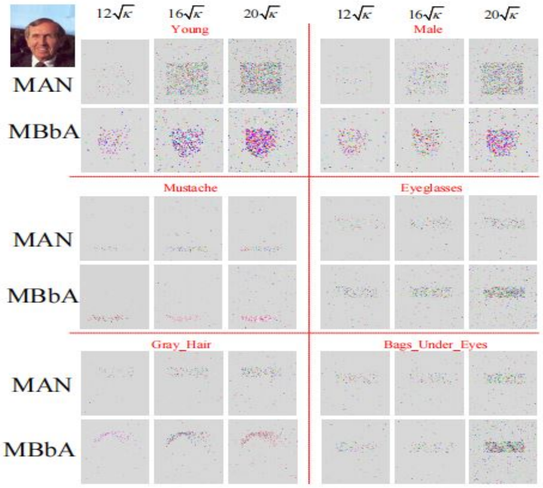


Figure 3: Distribution of adversarial perturbances with MAN and MBbA under different attack strength.

4.4 Ablation Study

Attack Strength c**:** Here we are still not very clear why MBbA can obtain satisfactory attack performance, the main reason is whether MBbA can effectively seek appropriate perturbance areas, and then generate the corresponding adversarial perturbances, so as to perform effective attacks? To resolve this confusion, we consider the $\text{MBbA}_{\text{SI}}^{D}$ scenario with the most attacked models, and all settings are the same as those in **Section** 4.2.2. We just choose the MAN as the comparison method because it achieves the best performance among the comparison algorithms. By changing the attack strength ($12\sqrt{\kappa}$, $16\sqrt{\kappa}$, and $20\sqrt{\kappa}$), we will observe the variations in adversarial perturbances. We use pre-trained MAN and MBbA under different attack strength to generate the corresponding adversarial examples with a randomly selected face image from CelebA, and then calculate the pixel differences between the original image and each adversarial example. Finally, we visualize these results with OpenCV [20] in Fig. 3.

It can be seen from Fig. 3 that as the attack strength increases, the adversarial perturbances become more and more obvious. In addition, we can see that the areas where MBbA and MAN generate dense perturbances are all related to the input targets, even if other regions are disturbed. More importantly, the perturbances generated by MBbA is closer to the target area than those by MAN. Therefore, we can conclude that when the input is the target category, both MBbA and MAN can capture the associated areas for interference. The unique structure of MBbA ensures that it can exploit those areas accurately, even for multiple attacked models. This is why MBbA can generate the most effective adversarial examples among all comparative methods.

5 CONCLUSION

In this paper, we first proposed an end-to-end black-box attack method (MBbA) to attack multiple models at the same time. By encoding the target categories and the input images into associated spaces, MBbA tries to exploit appropriate attack areas from the input images during training, and then conducts effective attacks. Compared with state-of-the-art methods: (1) MBbA not only achieves the best performance in a single black-box attack scenario, but also takes the least time to carry out the most effective attacks toward multiple black-box attacks; (2) the success rates of MBbA attacking multiple models simultaneously are the best; (3) the adversarial samples generated by MBbA show good transferability and can effectively improve the robustness of the attacked models. More importantly, the whole process takes the least amount of time. In future work, we will try to add the weight regularization term to reduce the overfitting of the system, thereby enhancing its generalization ability. Furthermore, we will consider more application scenarios, such as enhancing the defensiveness or testing the robustness of one system.

ACKNOWLEDGMENTS

This work was supported in part by the National Key-Research and Development Program of China under Grant No. 2020YFB2104003, in part by the National Outstanding Youth Science Program of National Natural Science Foundation of China under Grant 61625202, in part by the International Cooperation and Exchange Key Program of National Natural Science Foundation of China under Grant 61860206011, in part by the National Youth Science Program of National Natural Science Foundation of China under Grant 61902119, in part by the Open Fund of Science and Technology on Parallel and Distributed Processing Laboratory under Grant 6142110200205, in part by the Shenzhen Excellent Technological and Innovative Talent Training Foundation under Grant RCBS20200714114941176, and in part by the Project funded by China Postdoctoral Science Foundation under Grants 2019M652758 and 2019TQ0087. This paper is funded by the Hong Kong Scholars Program under Grants XJ2020032.

REFERENCES

[1] Tao Bai, Jun Zhao, Jinlin Zhu, Shoudong Han, Jiefeng Chen, and Bo Li. 2020. AI-GAN: Attack-Inspired Generation of Adversarial Examples. *arXiv preprint arXiv:2002.02196* (2020).

[2] Arjun Nitin Bhagoji, Warren He, Bo Li, and Dawn Song. 2018. Practical black-box attacks on deep neural networks using efficient query mechanisms. In *ECCV*. Springer, 158–174.

[3] Wieland Brendel, Jonas Rauber, and Matthias Bethge. 2018. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *ICLR* (2018).

[4] Nicholas Carlini and David Wagner. 2017. Towards evaluating the robustness of neural networks. In *SP*. IEEE, 39–57.

[5] Jianbo Chen, Michael I Jordan, and Martin J Wainwright. 2020. Hopskipjumpattack: A query-efficient decision-based attack. In *sp*. IEEE, 1277–1294.

[6] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. 2017. 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*. 15–26.

[7] Weilun Chen, Zhaoxiang Zhang, Xiaolin Hu, and Baoyuan Wu. 2020. Boosting decision-based black-box adversarial attacks with random sign flip. In *ECCV*.

[8] Qi Chu, Wanli Ouyang, Hongsheng Li, Xiaogang Wang, Bin Liu, and Nenghai Yu. 2017. Online multi-object tracking using CNN-based single object tracker with spatial-temporal attention mechanism. In *CVPR*. 4836–4845.

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*. 248–255.

[10] Ensheng Dong, Hongru Du, and Lauren Gardner. 2020. An interactive web-based dashboard to track COVID-19 in real time. *The Lancet infectious diseases* 20, 5 (2020), 533–534.

[11] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. 2018. Boosting adversarial attacks with momentum. In *CVPR*. 9185–9193.

[12] Yinpeng Dong, Hang Su, Baoyuan Wu, Zhifeng Li, Wei Liu, Tong Zhang, and Jun Zhu. 2019. Efficient decision-based black-box adversarial attacks on face recognition. In *CVPR*. 7714–7722.

[13] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. *ICLR* (2015).

[14] Yiwen Guo, Ziang Yan, and Changshui Zhang. 2019. Subspace Attack: Exploiting Promising Subspaces for Query-Efficient Black-box Attacks. In *NeurIPS*. 3825–3834.

[15] Jiangfan Han, Xiaoyi Dong, Ruimao Zhang, Dongdong Chen, Weiming Zhang, Nenghai Yu, Ping Luo, and Xiaogang Wang. 2019. Once a man: Towards multi-target attack via learning multi-target adversarial network once. In *CVPR*. 5158–5167.

[16] Emily M Hand and Rama Chellappa. 2017. Attributes for improved attributes: A multi-task network utilizing implicit and explicit relationships for facial attribute classification. In *AAAI*. 4068–4074.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*. 770–778.

[18] Julia Hirschberg and Christopher D Manning. 2015. Advances in natural language processing. *Science* 349, 6245 (2015), 261–266.

[19] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. 2018. Black-box Adversarial Attacks with Limited Queries and Information. In *ICML*. 2137–2146.

[20] Adrian Kaehler and Gary Bradski. 2016. *Learning OpenCV 3: computer vision in C++ with the OpenCV library*. "O'Reilly Media, Inc.".

[21] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436–444.

[22] Yandong Li, Lijun Li, Liqiang Wang, Tong Zhang, and Boqing Gong. 2019. NATTACK: Learning the Distributions of Adversarial Examples for an Improved Black-Box Attack on Deep Neural Networks. In *ICML*. 3866–3876.

[23] Fan Liu, Shuyu Zhao, Xuelong Dai, and Bin Xiao. 2021. Long-term Cross Adversarial Training: A Robust Meta-learning Method for Few-shot Classification Tasks. *In Proceedings of the ICML 2021 Workshop on Adversarial Machine Learning* (2021).

[24] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. 2016. Delving into transferable adversarial examples and black-box attacks. *ICLR* (2016).

[25] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep Learning Face Attributes in the Wild. In *ICCV*.

[26] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. 2016. Deepfool: a simple and accurate method to fool deep neural networks. In *CVPR*. 2574–2582.

[27] Muzammal Naseer, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Fatih Porikli. 2020. A Self-supervised Approach for Adversarial Robustness. In *CVPR*. 262–271.

[28] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. 2016. The limitations of deep learning in adversarial settings. In *EuroS&P*. IEEE, 372–387.

[29] Salil Prabhakar, Sharath Pankanti, and Anil K Jain. 2003. Biometric recognition: Security and privacy concerns. *SP* 1, 2 (2003), 33–42.

[30] Karl Ricanek and Tamirat Tesafaye. 2006. Morph: A longitudinal image database of normal adult age-progression. In *FGR*. IEEE, 341–345.

[31] Rasmus Rothe, Radu Timofte, and Luc Van Gool. 2018. Deep expectation of real and apparent age from a single image without facial landmarks. *IJCV* 126, 2-4 (2018), 144–157.

[32] Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. *ICLR* (2015).

[33] Yi Sun, Yuheng Chen, Xiaogang Wang, and Xiaoou Tang. 2014. Deep learning face representation by joint identification-verification. In *NeurIPS*. 1988–1996.

[34] Fnu Suya, Jianfeng Chi, David Evans, and Yuan Tian. 2020. Hybrid batch attacks: Finding black-box adversarial examples with limited queries. In *29th USENIX Security Symposium (USENIX Security 20)*. 1327–1344.

[35] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *ICLR* (2013).

[36] Chun-Chen Tu, Paishun Ting, Pin-Yu Chen, Sijia Liu, Huan Zhang, Jinfeng Yi, Cho-Jui Hsieh, and Shin-Ming Cheng. 2019. Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. In *AAAI*, Vol. 33. 742–749.

[37] Hongjun Wang, Guangrun Wang, Ya Li, Dongyu Zhang, and Liang Lin. 2020. Transferable, Controllable, and Inconspicuous Adversarial Attacks on Person Re-identification With Deep Mis-Ranking. In *CVPR*. 342–351.

[38] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. 2018. Generating adversarial examples with adversarial networks. In *IJCAI*. 3905–3911.

[39] Mingyi Zhou, Jing Wu, Yipeng Liu, Shuaicheng Liu, and Ce Zhu. 2020. DaST: Data-free Substitute Training for Adversarial Attacks. In *CVPR*. 234–243.

[40] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *CVPR*. 2223–2232.