MGAAttack: Toward More Query-efficient Black-box Attack by Microbial Genetic Algorithm

Lina Wang [1,2], Kang Yang[1,2], Wenqi Wang[1,2], Run Wang[3,†], Aoshuang Ye[1,2]
[1] Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, Wuhan, China
[2] School of Cyber Science and Engineering, Wuhan University, China
[3] Nanyang Technological University, Singapore
{lnwang, kang_yang, wangwenqi_001, wangrun, yasfrost}@whu.edu.cn

ABSTRACT

Recent studies have shown that deep neural networks (DNNs) are susceptible to adversarial attacks even in the black-box settings. However, previous studies on creating black-box based adversarial examples by merely solving the traditional continuous problem, which suffer query efficiency issues. To address the efficiency of querying in black-box attack, we propose a novel attack, called *MGAAttack*, which is a query-efficient and gradient-free black-box attack without obtaining any knowledge of the target model. In our approach, we leverage the advantages of both transfer-based and scored-based methods, two typical techniques in black-box attack, and solve a discretized problem by using a simple yet effective microbial genetic algorithm (MGA). Experimental results show that our approach dramatically reduces the number of queries on CIFAR-10 and ImageNet and significantly outperforms previous work. In the untargeted attack, we can attack a VGG19 classifier with only 16 queries and give an attack success rate more than 99.90% on ImageNet. Our code is available at https://github.com/kangyangWHU/MGAAttack.

CCS CONCEPTS

• **Security and privacy**; • **Computing methodologies** → **Computer vision**;

KEYWORDS

Deep neural networks, black-box adversarial attack, microbial genetic algorithm

ACM Reference Format:
Lina Wang, Kang Yang, Wenqi Wang, Run Wang, Aoshuang Ye. 2020. *MGAAttack*: Toward More Query-efficient Black-box Attack by Microbial Genetic Algorithm. In *Proceedings of the 28th ACM International Conference on Multimedia (MM '20), October 12–16, 2020, Seattle, WA, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3394171.3413703

1 INTRODUCTION

Deep learning has achieved significant progress in image classification [8], speech recognition [17], machine translation [3], face recognition [28], and object detection [29]. However, DNNs are easily fooled by adversarial examples [4, 37], which are crafted by adding some human imperceptible perturbations to benign inputs. Thus, serious security and privacy issues had raised in many critical fields [39, 40, 42] when deploying DNN-based systems. Therefore, adversarial example attacks and defenses is becoming the hottest research topics in the machine learning and cybersceurity communities. Many efforts are progressed in exploring the vulnerabilities of DNN models and building trustworthy DNN models which could be deployed in safety-critical applications.

In general, the adversarial attack can be divided into white-box attack and black-box attack based on the knowledge of target models that obtained by attackers. In the white-box setting, the attacker obtains the full-knowledge of the target model, thus attackers could utilizes the parameters of target model to achieve an success attack in high confidence. But white-box attack is not practical and most of time the target model is infeasible to attackers. In the more realistic black-box setting, the attacker can only obtain input and output pairs of the target model. The main approaches in black-box setting are transfer-based attacks [5, 9, 10, 12, 18, 23, 25, 43] and score-based attacks [1, 6, 20, 21, 27, 32, 33, 38]. The former transfer-based attack first generates candidate adversarial examples by applying a standard white-box attack method on the local model, and then leverages the transferability of the candidate adversarial examples to attack the target models. The latter score-based attack obtains the results of the input by querying the target model and then generates adversarial examples through gradient-based methods [6, 20, 21, 33, 38] or gradient-free methods [32].

The transfer-based attacks are very efficient, but the success rate is low. On the contrary, the scored-based attacks can achieve high success rate but it suffers from low query efficiency. Thus, recently some researchers [33] combine those two black-box attacks, using adversarial examples generated by transfer-based attacks as the starting point of the score-based attacks. Such attacks achieve high success rate and reduce the query times.

Most of the previous methods generate adversarial examples by solving the traditional continuous problem. However, Moon et al. [27] consider a discretized problem in which the perturbations are select from the vertices of the L_∞ ball. They significantly reduce the number of queries by solving this optimization problem. But they fail to leverage the information of the transfer-based attacks, which could further reduce the number of queries. Motivated by

[†] Run Wang is the corresponding author.

MM '20, October 12–16, 2020, Seattle, WA, USA
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-7988-5/20/10…$15.00
https://doi.org/10.1145/3394171.3413703

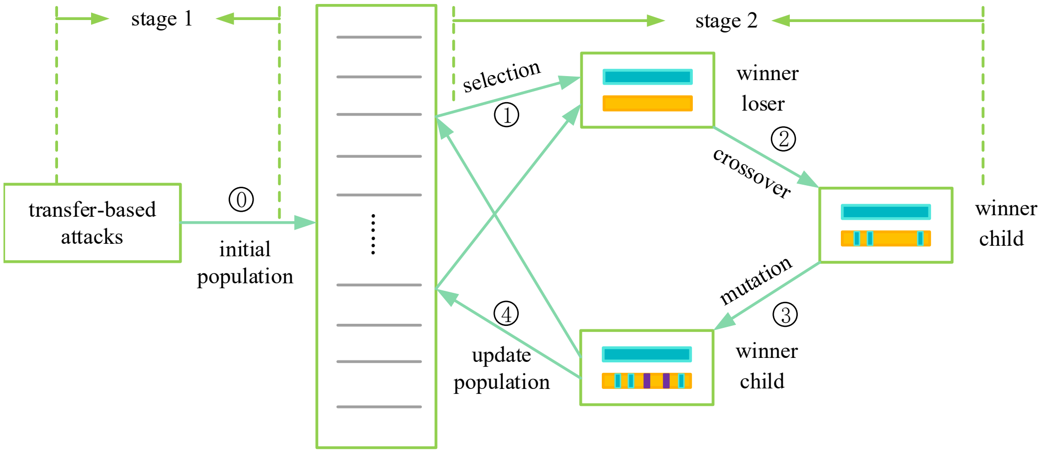


Figure 1: Framework of MGAAttack. MGAAttack contains two stages: (1) initial the population of MGA with adversarial examples generated by transfer-based attacks; (2) repeat selection, crossover, mutation, and update population until success.

their weakness, we combine transfer-based attacks and scored-based attacks to solve the discretized problem. On the one hand, the existing gradient estimation methods are not suitable for solving the discretized problem. On the other hand, the heuristic algorithm, such as genetic algorithm, evolution strategies, has been applied in optimization problems in recent years. In this work, we employ an alternative strategies by adopting MGA [15] to solve the discretized problem.

In this paper, we propose MGAAttack, a query-efficient black-box attack based on MGA. As illustrated in Figure 1, MGAAttack contains two stages:

(1) initial the population of MGA with adversarial examples generated by transfer-based attacks;
(2) repeat selection, crossover, mutation, and update population until success.

MGAAttack is a query-efficient attack thanks to the combination of transfer-based attacks and score-based attacks; Furthermore, MGAAttack is also a powerful attack with high attack success rate on many conventional models for image classification due to applying gradient-free MGA to solve the discretized problem.

The main contributions of our work are concluded as follows:

- We propose MGAAttack, a novel query-efficient and gradient-free black-box attack for crafting adversarial examples. It combines transfer-based attacks and score-based attacks and applies MGA to solve the discretized problem.
- Experimental results show that our method achieves competitive performance on CIFAR-10 and ImageNet. In untargeted setting, our method achieves nearly 100% success rate with query times less than 200, across undefended models on ImageNet. In targeted setting, our attacks can reach, 99.79% of success rate against VGG19 classifier within 1,680 queries on average on ImageNet.
- We demonstrate that MGAAttack is also robust against some typical defense mechanisms using gradient obfuscation [2], such as JPEG compression, bit depth reduction and random resizing and padding. Compared with previous studies on black-box attacks, MGAAttack can achieve higher or similar attack success rate, but our MGAAttack reduces the number of queries more than 37%.

2 RELATED WORK

In this section, we overview the recent work of black-box attacks which is divided into transfer-based attacks, score-based attacks, and combination attacks with transfer-based and score-based attacks.

2.1 Transfer-based attacks

Goodfellow et al. [37] observed that adversarial examples generated by one model are likely to fool other models with similar architectures. Therefore, Adversarial examples generated by white-box attacks on local models can be used to attack unknown models, which called transfer-based attacks. Standard white-box attacks, such as fast gradient sign method (FGSM) [12], basic iterative method (BIM) [23], Carlini-Wagner (CW) [5] can be used to perform transfer-based attacks, but they suffer from low success rate. Therefore, many efforts have been devoted to improve transferability. Dong et al. [9] propose an ensemble attack, which further improves the transferability through ensembling some different local models. Huang et al. [18] perturbs the hidden layer to enhance the transferability. Xie et al. [43] apply random and differentiable transformations to the input images before running white-box attacks to improve the transferability capabilities. Dong et al. [10] generate more transferable adversarial examples by convolving the gradient at the untranslated image with a pre-defined kernel.

2.2 Score-based attacks

Scored-based attacks obtain the input and output pairs by sending large amount of inputs to the target model. It can be group into gradient-based attacks and gradient-free attacks, depending on whether gradient information is used or not. The former leverages the scores of the models to estimate gradients. The latter applies gradient-free methods to generate adversarial examples.

Gradient-based attacks. Gradient-based black-box attacks first estimate the gradients by querying the target model and then apply them to run white-box attacks. ZOO [6] is the first gradient-based black-box attack. It adopts the finite difference method with dimension-wise estimation to approximate gradient values, and use them to perform a CW white-box attack. AutoZOOM [38] uses a vector-wise random gradient-free method to estimate gradients and uses an autoencoder or a simple bilinear resizing operation to reduce the attack dimension. Ilyas et al. [20] apply natural evolutional strategy (NES) to estimate gradients and generate adversarial examples through PGD [25] attack. Bandits [21] attack combine time and data priors with bandits framework and still apply NES to estimate the gradients. The gradient-based adversarial attack heavily relies on gradients, which suffers the gradient obfuscation defense mechanism. Hence, they are not practical in real attacks. Our proposed MGAAttack is a gradient-free attack that is robust against gradient obfuscation defenses (e.g., JPEG compression, bit depth reduction, and random resizing and padding) demonstrated in our experiments, thus poses a new threat to the community and calls for effective defense methods.

Gradient-free attacks. Gradient-free black-box attacks usually generate adversarial examples with heuristic methods, such as the genetic algorithm, and evolutional strategy. Su et al. [32] introduce one pixel attack that applies the differential evolution algorithm to perturb the important pixels of the image. Alzantot et al. [1] propose GenAttack, which generates adversarial examples through the genetic algorithm. Meunier et al. [26] explore a larger spectrum of evolution strategies, such as the covariance matrix adaptation evolution strategy (CMA-ES) [14], to generate adversarial examples. Moon et al. [27] propose the parsimonious attack, using combinational optimization to solve the discretized problem, which achieves SOTA performance.

2.3 Combination attacks

In addition to apply transfer-based or gradient-based attack alone, recenly some reseacher combine those two black-box attacks. Suya et al. [33] proposed Trans-AutoZOOM and Trans-NES, using adversarial examples generated from local models as starting points for AutoZOOM and NES. TREMBA [19] learned a low dimensional embedding using a pre-trained model and then applying NES to estimate the gradients in the low dimensional embedding. Du et al. [11] presented meta attack that training a meta attacker learns to estimate the gradient, then replace the zeroth-order gradient estimation in traditional black box attack methods with it to directly estimate the gradient.

Studies combining transfer-based and score-based solve a continuous optimization problem which suffers computing-consuming issues due to the large search space in finding adversarial inputs. On the contrary, we solve the discretized optimization problem, which can significantly reduce the search space and improve query efficiency. In addition, meta attack and TREMBA attack all need to train a local model to conduct black-box adversarial attack, thus it suffers efficiency issues when the task is complicated. Our proposed MGAAttack directly generates inputs to attack the target model without any training of a local attack model and is more efficient and general than these two attacks.

3 METHODS

In this section, we first introduce the two types of optimization problems, then detail our approach.

3.1 Two types of optimization problems

Considering a well-trained DNN classifier $F(x)$, where $x \in [0,1]^{\dim(x)}$ is the input of the network, the ground-truth label of the input x is y. We denote $F(x)_i$ as the i-th dimension of the network output. As an attacker, the goal is to find an input x_{adv}, that results in the change of class prediction while the distance between the adversarial and benign input is smaller than a predefined threshold ϵ as (1):

$$\arg\max_i F(x_{adv})_i \neq y, \quad s.t. \quad \|x_{adv} - x\|_p \leq \epsilon \tag{1}$$

where ϵ is the strength of distortion, the distance norm function L_p is often chosen as L_2 and L_∞. We focus on L_∞ in the following paper.

The continuous problem. Traditionally, adversarial examples can be generated by solving constrained continuous problem as follows:

$$x_{adv} = \arg\max_{x'} L(x', y), \quad s.t. \quad \left\|x' - x\right\|_\infty \leq \epsilon \tag{2}$$

where L is a loss function.

The discretized problem. Moon et al. [27] observed that PGD attack pushes the perturbations towards the corner of the L_∞ ball. Therefore, they consider a discrete surrogate problem as (3):

$$x_{adv} = \arg\max_{x'} L(x', y), \quad s.t. \quad \left\|x' - x\right\|_\infty \in \{-\epsilon, \epsilon\} \tag{3}$$

In this view, generating adversarial examples can be treated as a combinational optimization problem. Previous methods used in the continuous problem can not be directly applied in solving this problem. On the contrary, genetic algorithms are suitable for this optimization problem, so we apply MGA to craft adversarial examples.

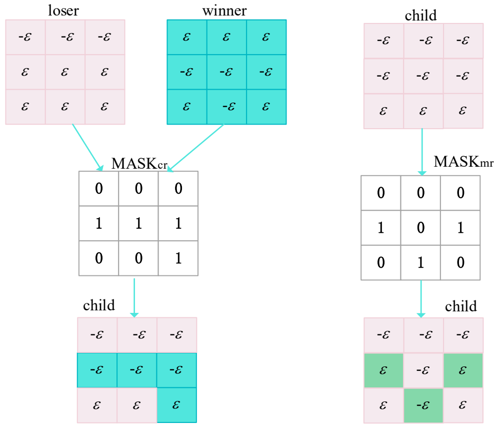
3.2 MGAAttack

MGAAttack relies on MGA, which is the population-based and gradient-free optimization algorithm. MGA is a variant of the Genetic algorithm, and more practical and straightforward. In MGA, a population of candidate solutions (called individuals) are iteratively evolved toward better solutions (larger fitness). The population in each iteration is called a generation. In each generation, the quality of population members is assessed using a fitness function. The fitness is usually the value of the objective function in the optimization problem being solved. The larger the fitness of the individuals is, the more likely it is to be selected for breeding the next generation. The next generation is generated through a combination of crossover and mutation. The details of MGAAttack in untargeted attack are given in Algorithm 1.

As described in Algorithm 1, MGAAttack includes the typical operators in genetic algorithm: *initialization* (line 1-2), *selection* (line 4-6), *crossover* (line 7), *mutation* (line 8), and *update population* (line 12).

Algorithm 1 MGAAttack (untargeted case).

```
Input: input $x$, true label $y$, distortion $\epsilon$, mutation rate $mr$,
    crossover rate $cr$, population size $N$, generation $G$, local model
    $M$, target model $F$
Output: adversarial example $x_{adv}$
$x'$ = transfer_based_attack($M$, $x$, $N$)
pop = init_population($x$, $x'$, $\epsilon$)
for $g = 1$ to $G$ do
    $p_1$, $p_2$ = random_select(pop)
    $f_1$, $f_2$= get_fitness($F$, $x$, $p_1$, $p_2$)
    loser, winner = sort_by_fitness( $p_1$, $p_2$, $f_1$, $f_2$)
    child = crossover($cr$, loser, winner)
    child = mutation($mr$, child)
    if $\arg\max_i F(child)_i \neq y$ then
        return child
    end if
    pop = update_population(pop, child)
end for
```



(a) crossover (b) mutation

Figure 2: The illustration of crossover operator (a) and mutation operator (b)

Initialization. The initial population is critical to the convergence of the algorithm. If the initial population is similar to the optimal solution, the algorithm will converge soon. Previous methods employ genetic algorithm initial population randomly. We use adversarial examples generated by transfer-based attacks to initialize the individual of population $\delta_i, i = \{1, 2, ..., N\}$ as follows

$$\delta_i = \begin{cases} -\epsilon & x_i' - x < 0 \\ \epsilon & x_i' - x \geq 0 \end{cases} \tag{4}$$

where x_i' is the adversarial examples generated by transfer-based attacks.

Fitness function. The fitness function is used to evaluate the quality of population members. It eventually leads MGA to evolve towards a population with larger fitness. As the fitness function should reflect the optimization objective, we use the loss function of equation (3) as the fitness function in untargeted setting.

Selection. Selection is used to decide who can pass on genetic information to the next generation. Unlike traditional genetic algorithm select the two parents by fitness proportionate selection. MGA randomly selects two individuals from the population. By comparing their fitnesses, we obtain a winner (larger fitness) and a loser.

Crossover. Crossover enables individuals with high fitness to pass on their genetic information to offspring. We have a winner and a loser after selection. We get an offspring by copying the genetic information of the winner and loser according to the crossover rate cr as follows:

$$\delta_{child} = \delta_{winer} * MASK_{cr} + \delta_{losser} * (1 - MASK_{cr}) \tag{5}$$

where

$$MASK_{cr} = \begin{cases} 1 & rand(0,1) < cr \\ 0 & otherwise \end{cases} \tag{6}$$

where $MASK_{cr}$ is a matrix of the same size of the individual, rand(0, 1) means uniformly generating a number between 0 and 1. cr represents the probabilities of crossover. δ_{loser} is the individual of loser, δ_{winner} is the individual of winner. Figure 2 (a) illustrates the crossover operator.

Mutation. Mutation can keep population diversity and avoid population trapping into local optima. We can easily carry out bit mutation on the child due to binary encoding as (7):

$$\delta_{child} = -\delta_{child} * MASK_{mr} + \delta_{child} * (1 - MASK_{mr}) \tag{7}$$

where

$$MASK_{mr} = \begin{cases} 1 & rand(0,1) < mr \\ 0 & otherwise \end{cases} \tag{8}$$

where $MASK_{mr}$ is a matrix of the same size of the individual, δ_{child} is the offspring generated by crossover, mr represents the probabilities of mutation. The illustration of the mutation operator is shown in Figure 2 (b).

Update population. Update population makes the population evolve continuously. We replace the loser with the offspring derived from the loser and keep the winner unchanged.

In summary, after initializing the population, MGAAttack picks two individuals at random, and we have a winner and a loser by comparing their fitnesses. Then we generate a new offspring from the loser by crossover and mutation. Finally, we replace the loser by the offspring. MGAAttack repeats these steps until successed. Our framework is illustrated in Figure 1.

4 EXPERIMENTS

We evaluated the performance of MGAAttack against the vanilla networks and the robustness against defensive models on two datasets, CIFAR-10 [22] and ImageNet [30]. Additionally, we measure the sensitivity of MGAAttack to hyperparameters. We only consider L_∞ threat model and quantify the performance with attack success rate, average queries, and median queries. We compare our attack with NES [20], Trans-NES [33], Bandits [21], and parsimonious [27], which are SOTA black-box attacks. We use the original code for Bandits and parsimonious, and port the code from TensorFlow to PyTorch for NES and Trans-NES in our experiments. All parameters keep aligning with those recommended by papers. We

Table 1: Success rate, average queries, and median queries of untargeted attack on ImageNet.

Method	VGG19			Resnet50			InceptionV3		
	Success Rate	Avg. Queries	Med. Queries	Success Rate	Avg. Queries	Med. Queries	Success Rate	Avg. Queries	Med. Queries
NES	87.60%	1079	450	81.77%	1433	650	67.22%	1694	600
Trans-NES	98.35%	78	50	91.76%	163	50	86.29%	209	50
Bandits	94.21%	298	56	95.57%	735	218	**98.43%**	488	42
Parsimonious	98.86%	254	129	99.49%	247	131	97.93%	863	306
Ours	**99.90%**	**16**	**5**	**99.90%**	**100**	**5**	97.21%	**175**	**5**

Table 2: Success rate, average queries, and median queries of targeted attack on ImageNet.

Method	VGG19			Resnet50			InceptionV3		
	Success Rate	Avg. Queries	Med. Queries	Success Rate	Avg. Queries	Med. Queries	Success Rate	Avg. Queries	Med. Queries
NES	99.79%	9566	8075	**99.90%**	10387	8975	96.49%	15853	13650
Trans-NES	99.79%	6325	4580	**99.90%**	7755	6350	97.00%	14503	11950
Bandits	90.29%	14405	10794	90.83%	15555	13045	51.71%	22499	20954
Parsimonious	**99.90%**	3190	2464	**99.90%**	**2803**	2190	**99.48%**	8577	6215
Ours	99.79%	**1680**	**750**	95.59%	2827	**1545**	96.70%	**7973**	**4233**

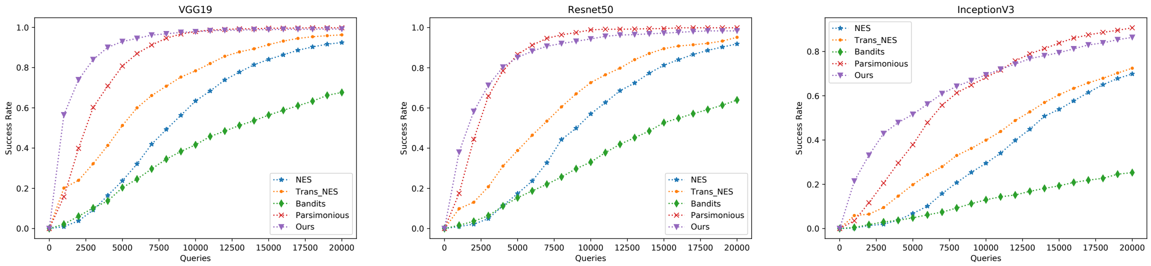


Figure 3: The success rate of targeted attack at different query levels for different models on ImageNet.

only run Bandits attack on ImageNet, as it does not fit on CIFAR-10. Parsimonious attack is the strongest attack among the four attacks, but it does not work on MNIST [24], so we do not run experiments on this dataset.

Hyperparameters. In all experiments, we set the *cr* to 0.7 and N to 5. We take ensemble MI-FGSM with iterations of 10 and a step size of 0.01 as the transfer-based attack. For CIFAR-10, we set ϵ to 8 in [0,255] scale, and the maximum queries to 5000 and 10000 in untargeted setting and targeted setting, respectively. For ImageNet, we set ϵ to 12 in [0,255] scale, and the maximum queries to 10000 and 50000 in untargeted setting and targeted setting, respectively. In targeted setting, targeted classes were chosen randomly for each image, and all attacks chose the same target class for the same image for a fair comparison.

4.1 Black-box attacks on ImageNet

We randomly chose 1,000 correctly classified images from ImageNet validation set for evaluating the attacks. We use the VGG16 [31], Resnet18 [16], and Resnet34 [16] as the local models when attacking VGG19 and Resnet50. We use the InceptionV4 [34], Xception [7], and Inception-resnetv2 [34] as the local models when attacking InceptionV3 [36]. We evaluated all attacks on VGG19 [31], Resnet50 [16], and InceptionV3 [36]. We set the *mr* to 1e-3 and 3e-4 for untargeted setting and targeted setting, respectively.

We report the results of untargeted attacks and targeted attacks in Table 1 and Table 2, respectively. In untargeted setting, we achieve a nearly 100% success rate within 100 queries on two models. To our best knowledge, this is the SOTA result. In targeted setting, our method achieves comparable success rate and average queries with lower median queries compared to the strongest attack. Figure 3 plots the success rate under different query levels, we find that our method converges quickly. Our method still can achieve about 80% success rate on VGG19 and Resnet50 model when the number of queries are limited to 2,500.

4.2 Black-box attacks on CIAFR-10

In this section, we evaluate our attack on CIFAR-10 for both untargeted setting and targeted setting. We use the pre-trained models on CIFAR-10 provided by huyvnphan[1], which are the SOTA models. We use VGG16 with batch normalization [31], Googlenet [35], and Resnet18 as the local models. We set the mutation rate *mr* to 1e-3 for both settings.

Table 3 and Table 4 list the untargeted attack and targeted attack results on VGG19 with batch normalization [31], inceptionV3, and

[1] https://github.com/huyvnphan/PyTorch-CIFAR10

Table 3: Success rate, average queries, and median queries of untargeted attack on CIFAR-10.

Method	VGG19_bn			Resnet50			InceptionV3		
	Success Rate	Avg. Queries	Med. Queries	Success Rate	Avg. Queries	Med. Queries	Success Rate	Avg. Queries	Med. Queries
NES	80.50%	655	350	54.11%	675	400	80.76%	529	300
Trans-NES	79.95%	302	50	59.62%	533	250	86.22%	227	50
Trans-Rand	73.4%	100	**1**	39.64%	**191**	**2**	78.63%	**97**	**1**
Parsimonious	99.78%	522	327	**95.83%**	862	389	98.64%	562	296
Ours	**99.94%**	**130**	5	95.34%	551	151	**99.72%**	**98**	5

Table 4: Success rate, average queries, and median queries of targeted attack on CIFAR-10.

Method	VGG19_bn			Resnet50			InceptionV3		
	Success Rate	Avg. Queries	Med. Queries	Success Rate	Avg. Queries	Med. Queries	Success Rate	Avg. Queries	Med. Queries
NES	74.16%	3288	2550	76.34%	3195	2400	93.87%	2113	1350
Trans-NES	78.40%	2463	1200	77.93%	2861	1950	96.04%	1598	800
Parsimonious	95.13%	2495	1596	**93.93%**	2425	1467	98.05%	1602	982
Ours	**95.88%**	**1157**	**350**	91.18%	**1879**	**1011**	**99.12%**	**686**	**250**

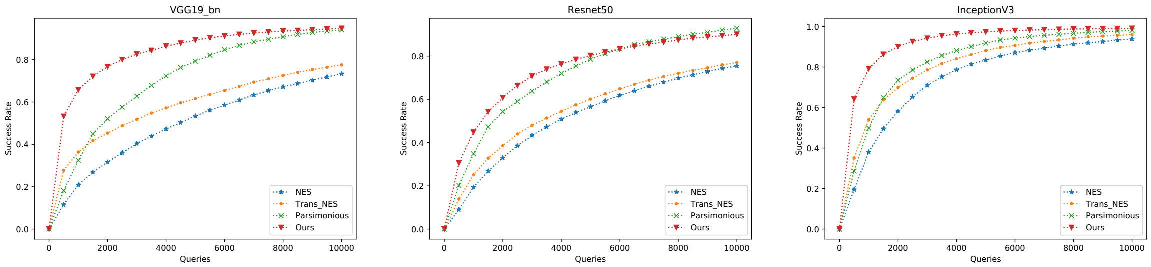


Figure 4: Success rate of targeted attack at different query levels for different models on CIFAR-10.

Resnet50, respectively. Trans-Rand generates adversarial examples by randomly flipping some perturbations of candidate adversarial examples. The median of Trans-Rand is very low, indicating that the candidate adversarial examples are very close to the final adversarial examples in the discretized problem. Therefore, random operation can achieve well performance. It well explains why MGA can achieve SOTA performance. Our methods achieve higher success rate with more than 50% reduction of queries compared with other attacks. Figure 4 plots the success rate of targeted attack under different query levels, our method converges quickly and achieves higher success rate at all query levels.

4.3 Black-box attacks against defensive models on ImageNet

In this section, we validate the effectiveness of our proposed method by attacking several defensive models on ImageNet, including JPEG compression [13], random resizing and padding (R&P) [41]. For JPEG compression, the quality level was set to 75, while for R&P, we keep aline with the setting of paper [41]. The experimental setting are the same with those untargeted attacking the normal InceptionV3 model in section 4.1. As shown in section 4.1, the performance of bandits attack is only comparable to NES attack, so we just ignore bandits attack in this section.

As shown in Table 5, even the two defensive methods are very simple, they do migrate the adversarial attacks in most case, especial R&P. We also notice that the success rate of parsimonious drops down to 37.42% when attacking R&P defensed models. On the contrary, our method performs well against two defensive methods.

4.4 Black-box attacks against defensive models on CIFAR-10

In this section, we evaluate the effectiveness of our method by attacking several defensive models on the CIFAR-10, including adversarial training [25], JPEG compression [13], and bit depth reduction [13]. We adversarially trained a WideRestnet34x10 [44] with a PGD L_∞ attack. The model achieves a 79.31% clean accuracy and a 49.44% accuracy under PGD L_∞ attacks with a norm of 8/255. We use VGG19 with batch normalization as the target model for the JPEG compression and bit depth reduction. For JPEG compression, the quality level was set to 75, while for bit depth reduction, the last 5 bits are reduced, as done in [13]. We run all attacks on all test images, so we restrict the maximum number of queries to 10,000. We set the mutation rate mr to 5e-4 for adversarial training, and 1e-3 for the other two defensive models. The results are presented in Table 6.

Table 5: Results of untargeted attack against 2 defensive models on ImageNet.

Method	Vanilla			JPEG			R&P		
	Success Rate	Avg. Queries	Med. Queries	Success Rate	Avg. Queries	Med. Queries	Success Rate	Avg. Queries	Med. Queries
NES	67.22%	1694	600	18.54%	1650	500	34.75%	1964	900
Trans-NES	86.29%	209	50	71.98%	78	50	80.21%	89	50
Parsimonious	97.93%	863	306	**98.13%**	765	296	37.42%	1556	321
Ours	97.21%	**175**	**5**	92.92%	154	**5**	**89.70%**	**69**	**5**

Table 6: Results of untargeted attack against 3 defensive models on CIFAR-10.

Method	Adversarial Training			JPEG Compression			Bit-depth Reduction		
	Success Rate	Avg. Queries	Med. Queries	Success Rate	Avg. Queries	Med. Queries	Success Rate	Avg. Queries	Med. Queries
NES	15.31%	995	250	28.97%	949	350	57.24%	1210	350
Trans-NES	29.55%	**52**	50	35.47%	450	50	60.01%	592	50
Parsimonious	**38.73%**	939	335	93.37%	743	237	99.78%	350	195
Ours	36.59%	593	**5**	**99.85%**	**163**	**5**	**99.87%**	**164**	**5**

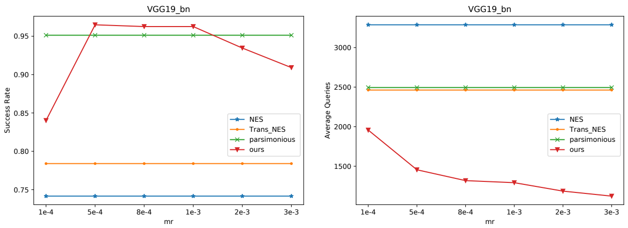


Figure 5: The success rate (left) and average queries (right) at different *mr* targeting against VGG19_bn on CIFAR-10.

Although our method achieves slightly lower success rate than the parsimonious attack when attacking adversarially trained models, our method achieves about 37% reduction of queries compare to it. What's more, our attack can defeat non-differentiable input transformation with a remarkable margin. Compared with the other attacks, our method achieves higher success rate and reduces more than 50% of queries against the other two defensive models.

Through attacking the defensive models on CIFAR-10 and ImageNet, we find that our method and Trans-NES are more powerful against those defensive methods performing gradient obfuscation, are less effective against adversarially trained models. We think both our method and Trans-NES use the adversarial examples generated by transfer-based attacks as the starting point, adversarial examples are harder to transfer between adversarially trained models.

4.5 Ablation Study

In this section, we investigate the influence of the mutation rate to the black-box attack effect. We run those experiments on CIAFR-10 in targeted setting. We use VGG19 with batch normalization as the targeted model. We change the mutation rate and keep the others the same as in section 4.2. Figure 5 plots the success rate and average queries under different mutation rates respectively. We find that although MGAAttack is sensitive to mutation rate, it can perform well in a certain range. We can quickly find the range due to the following two findings:

(1) The mutation rate is related to image dimension and attack type, and we only need to adjust once for a setting. In general, the more challenging to succeed, the smaller the mutation rate is.

(2) In general, the mutation rate is greater than 1e-4 and no more than 1e-2.

5 CONCLUSION

In this paper, we present a query-efficient black-box attack, MGAAttack, which constructs adversarial examples with the MGA algorithm. MGAAttack contains two stages: (1) initial the population of MGA with adversarial examples generated by transfer-based attacks; (2) repeat selection, crossover, mutation, and update population until success. Experimental results show that our method is query-efficient and powerful against vanilla models by evaluating on CIFAR-10 and ImageNet. Our method outperforms Trans-NES in all experiments, even though both methods leverage the information of transfer-based attacks. Furthermore, through attacking defensive models on CIFAR10 and ImageNet, we find that our method is more powerful against those defensive methods performing gradient obfuscation, is less effective against adversarially trained models. The reason is that our method partly dependent on transfer-based attacks. Finally, we measure the robustness of our method to hyperparameters, and we show our method is only sensitive to mr, which is also easy to adjust according to our findings. Consequently, MGAAttack provides an efficient and powerful way to access the robustness of models.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (NSFC) Under Grants No.61876134, No.U1836112, and No.U1536204.

REFERENCES

[1] Moustafa Alzantot, Yash Sharma, Supriyo Chakraborty, Huan Zhang, Cho-Jui Hsieh, and Mani B Srivastava. 2019. Genattack: Practical black-box attacks with gradient-free optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 1111–1119.

[2] Anish Athalye, Nicholas Carlini, and David Wagner. 2018. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *35th International Conference on Machine Learning, ICML 2018* 1 (2018), 436–448.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[4] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. 2013. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*. Springer, 387–402.

[5] Nicholas Carlini and David Wagner. 2017. Towards evaluating the robustness of neural networks. In *2017 IEEE symposium on security and privacy (sp)*. 39–57.

[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] François Chollet. 2017. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1251–1258.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. IEEE, 248–255.

[9] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. 2018. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 9185–9193.

[10] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. 2019. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4312–4321.

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

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

[13] Chuan Guo, Mayank Rana, Moustapha Cissé, and Laurens Van Der Maaten. 2018. Countering adversarial images using input transformations. In *International Conference on Learning Representations, ICLR 2018*. 1–12.

[14] Nikolaus Hansen and Andreas Ostermeier. 2001. Completely derandomized self-adaptation in evolution strategies. *Evolutionary computation* 9, 2 (2001), 159–195.

[15] Inman Harvey. 2009. The microbial genetic algorithm. In *European conference on artificial life*. Springer, 126–133.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[17] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine* 29, 6 (2012), 82–97.

[18] Qian Huang, Isay Katsman, Horace He, Zeqi Gu, Serge Belongie, and Ser-Nam Lim. 2019. Enhancing adversarial example transferability with an intermediate level attack. In *Proceedings of the IEEE International Conference on Computer Vision*. 4733–4742.

[19] Zhichao Huang and Tong Zhang. 2019. Black-Box Adversarial Attack with Transferable Model-based Embedding. *arXiv preprint arXiv:1911.07140* (2019).

[20] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. 2018. Black-box adversarial attacks with limited queries and information. *arXiv preprint arXiv:1804.08598* (2018).

[21] Andrew Ilyas, Logan Engstrom, and Aleksander Madry. 2018. Prior convictions: Black-box adversarial attacks with bandits and priors. *arXiv preprint arXiv:1807.07978* (2018).

[22] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).

[23] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2016. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533* (2016).

[24] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

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

[26] Laurent Meunier, Jamal Atif, and Olivier Teytaud. 2019. Yet another but more efficient black-box adversarial attack: tiling and evolution strategies. *arXiv preprint arXiv:1910.02244* (2019).

[27] Seungyong Moon, Gaon An, and Hyun Oh Song. 2019. Parsimonious black-box adversarial attacks via efficient combinatorial optimization. *arXiv preprint arXiv:1905.06635* (2019).

[28] Omkar M Parkhi, Andrea Vedaldi, and Andrew Zisserman. 2015. Deep face recognition. (2015).

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

[30] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision* 115, 3 (2015), 211–252.

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

[32] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. 2019. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation* 23, 5 (2019), 828–841.

[33] Fnu Suya, Jianfeng Chi, David Evans, and Yuan Tian. 2019. Hybrid batch attacks: Finding black-box adversarial examples with limited queries. *arXiv preprint arXiv:1908.07000* (2019).

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

[35] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1–9.

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

[37] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199* (2013).

[38] 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 *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 742–749.

[39] Wenqi Wang, Lina Wang, Run Wang, Zhibo Wang, and Aoshuang Ye. 2019. Towards a Robust Deep Neural Network in Texts: A Survey. *arXiv preprint arXiv:1902.07285* (2019).

[40] Zhipeng Wei, Jingjing Chen, Xingxing Wei, Linxi Jiang, Tat-Seng Chua, Fengfeng Zhou, and Yu-Gang Jiang. 2020. Heuristic Black-Box Adversarial Attacks on Video Recognition Models.. In *AAAI*. 12338–12345.

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

[42] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan Yuille. 2017. Adversarial examples for semantic segmentation and object detection. In *Proceedings of the IEEE International Conference on Computer Vision*. 1369–1378.

[43] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L Yuille. 2019. Improving transferability of adversarial examples with input diversity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2730–2739.

[44] Sergey Zagoruyko and Nikos Komodakis. 2016. Wide residual networks. *arXiv preprint arXiv:1605.07146* (2016).