

# GenAttack: Practical Black-box Attacks with Gradient-Free Optimization

Moustafa Alzantot  
UCLA  
Los Angeles, U.S.A  
malzantot@ucla.edu

Huan Zhang  
UCLA  
Los Angeles, U.S.A  
huanzhang@ucla.edu

Yash Sharma  
Cooper Union  
New York, U.S.A  
sharma2@cooper.edu

Cho-Jui Hsieh  
UCLA  
Los Angeles, U.S.A  
chohsieh@cs.ucla.edu

Supriyo Chakraborty  
IBM Research  
New York, U.S.A  
supriyo@us.ibm.com

Mani B. Srivastava  
UCLA  
Los Angeles, U.S.A  
mbs@ucla.edu

## ABSTRACT

Deep neural networks are vulnerable to adversarial examples, even in the black-box setting, where the attacker is restricted solely to query access. Existing black-box approaches to generating adversarial examples typically require a significant number of queries, either for training a substitute network or performing gradient estimation. We introduce GenAttack, a gradient-free optimization technique that uses genetic algorithms for synthesizing adversarial examples in the black-box setting. Our experiments on different datasets (MNIST, CIFAR-10, and ImageNet) show that GenAttack can successfully generate visually imperceptible adversarial examples against state-of-the-art image recognition models with orders of magnitude fewer queries than previous approaches. Against MNIST and CIFAR-10 models, GenAttack required roughly 2,126 and 2,568 times fewer queries respectively, than ZOO, the prior state-of-the-art black-box attack. In order to scale up the attack to large-scale high-dimensional ImageNet models, we perform a series of optimizations that further improve the query efficiency of our attack leading to 237 times fewer queries against the Inception-v3 model than ZOO. Furthermore, we show that GenAttack can successfully attack some state-of-the-art ImageNet defenses, including ensemble adversarial training and non-differentiable or randomized input transformations. Our results suggest that evolutionary algorithms open up a promising area of research into effective black-box attacks.

## CCS CONCEPTS

• **Theory of computation** → **Evolutionary algorithms**; • **Computing methodologies** → **Neural networks**; *Computer vision*;

## KEYWORDS

Adversarial Examples, Deep Learning, Genetic Algorithm, Computer Vision

## ACM Reference Format:

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 *Genetic and Evolutionary Computation Conference (GECCO '19)*, July 13–17, 2019, Prague, Czech Republic. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3321707.3321749>

## 1 INTRODUCTION

Deep neural networks (DNNs) have achieved state-of-the-art performance in various tasks in machine learning and artificial intelligence. Despite their effectiveness, recent studies have illustrated the vulnerability of DNNs to adversarial examples [11, 29]. For instance, a virtually imperceptible perturbation to an image can lead a well-trained DNN to misclassify. Targeted adversarial examples can even cause misclassification to a chosen class. Moreover, researchers have shown that these adversarial examples are still effective in the physical world [4, 16], and can be crafted in distinct data modalities, such as in the natural language [2], and speech [1] domains. The lack of robustness exhibited by DNNs to adversarial examples has raised serious concerns for security-critical applications.

Nearly all previous work on adversarial attacks, [7, 8, 11, 12, 16, 21] has used gradient-based optimization in order to find successful adversarial examples. However, gradient computation can only be performed when the attacker has full knowledge of the model architecture and weights. Thus, these methods are only applicable in the *white-box* setting, where an attacker is given full access and control over a targeted DNN. However, when attacking real-world systems, one needs to consider the problem of performing adversarial attacks in the *black-box* setting, where nothing is revealed about the network architecture, parameters, or training data. In such a case, the attacker only has access to the input-output pairs of the classifier. A popular approach in this setting have relied on attacking trained substitute networks, and hoping the generated examples transfer to the target model [22]. This approach suffers from the inherent model mismatch between the substitute model to the target model, as well as the high computational cost required to train the substitute network. Recent works have used coordinate-based finite difference methods in order to directly estimate the gradients from the confidence scores, however the attacks are still computationally expensive, relying on optimization tricks to remain tractable [9].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).  
GECCO '19, July 13–17, 2019, Prague, Czech Republic  
© 2019 Association for Computing Machinery.  
ACM ISBN 978-1-4503-6111-8/19/07...\$15.00  
<https://doi.org/10.1145/3321707.3321749>

Both approaches are query-intensive, thus limiting their practicality in real-world scenarios.

Motivated by the above, we present *GenAttack*, a novel approach to generating adversarial examples without having to compute or even approximate the gradients, enabling efficient adversarial attacks to the black-box case. In order to perform *gradient-free optimization*, we adopt a population-based approach using genetic algorithms, iteratively evolving a population of feasible solutions until success. We also present a number of tricks that allows *GenAttack* to maintain its query-efficiency when attacking models trained on large-scale higher-dimensional datasets, such as ImageNet [10].

Due to its gradient-free nature, *GenAttack* is robust to defenses which perform gradient masking or obfuscation [3]. Thus, unlike many current black-box attack approaches, *GenAttack* can efficiently craft perturbations in the black-box setting to bypass some recently proposed defenses which manipulate the gradients.

We evaluate *GenAttack* using state-of-the-art image classification models, and find that the algorithm is successful at performing *targeted* black-box attacks with significantly less queries than current approaches. In our MNIST, CIFAR-10, and ImageNet experiments, *GenAttack* required roughly 2, 126, 2, 568, and 237 times less queries than the current state-of-the-art black-box attack, respectively. Additionally, we also demonstrate the success of *GenAttack* against state-of-the-art ImageNet defenses, such as ensemble adversarial training [30], and randomized, non-differentiable input transformation defenses [13]. These results illustrate the power of *GenAttack*'s query efficiency and gradient-free nature.

In summary, we make the following contributions:

- We introduce *GenAttack*, a novel gradient-free approach for generating adversarial examples by leveraging population-based optimization. Our implementation is open-sourced<sup>1</sup> to promote further research in studying adversarial robustness.
- We show that in the restricted black-box setting, *GenAttack* using genetic optimization, as well as dimensionality reduction and adaptive parameter scaling, can generate adversarial examples which force state-of-the-art image classification models, trained on MNIST, CIFAR-10 and ImageNet, to misclassify examples to chosen target labels with significantly less queries than previous approaches.
- We further highlight the effectiveness of *GenAttack* by illustrating its success against a few state-of-the-art ImageNet defenses, namely ensemble adversarial training and randomized, non-differentiable input transformations. To the best of our knowledge, we are the first to present a successful black-box attack against these defenses.

The rest of this paper is organized as follows: Section 2 provides a summary of related work. Section 3 formally defines the threat model for our attack. Section 4 discusses the details of *GenAttack*. Section 5 describes our evaluation experiments and their results. Finally, Section 6 concludes the paper with a discussion on the utility of evolutionary algorithms for generating adversarial examples.

## 2 RELATED WORK

In what follows, we summarize recent approaches for generating adversarial examples, in both the white-box and black-box cases, as

<sup>1</sup>[https://github.com/nesl/adversarial\\_genattack](https://github.com/nesl/adversarial_genattack)

well as defending against adversarial examples. Please refer to the cited works for further details.

### 2.1 White-box attacks & Transferability

In the *white-box* case, attackers have complete knowledge of and full access to the targeted DNN. In this scenario, the adversary is able to use backpropagation for gradient computation, which enables efficient gradient-based attacks. We briefly summarize a few important white-box attacks formulations below.

***L-BFGS***. [29] used box-constrained L-BFGS to minimize the  $\ell_2$  norm of the added adversarial noise  $\|\delta\|_2$  subject to  $f(x + \delta) = l$  (prediction is class  $l$ ) and  $x + \delta \in [0, 1]^m$  (input is within the valid pixel range), where  $f : \mathbb{R}^m \rightarrow \{1, \dots, k\}$  is the classifier, mapping a data example to a discrete label,  $l \in \{1, \dots, k\}$  is the target output label, and  $\delta$  is the added noise.

***FGSM & I-FGSM***. [11] proposed the Fast Gradient Sign Method (FGSM) to quickly generate adversarial examples. Under an  $L_\infty$  distortion constraint  $\|\delta\|_\infty \leq \epsilon$ , FGSM uses the sign of the gradient of the training loss  $J$  with respect to the original  $x_0$  and the true label  $l$ , to generate an adversarial example:  $x = x_0 + \epsilon \cdot \text{sign}(\nabla J(x_0, l))$ . Similarly, targeted attacks can be implemented by computing the loss with respect to a specified target class  $t$ , and instead going in the direction of the negative gradient.

In [16], an iterative version of FGSM was proposed (I-FGSM), where FGSM is used iteratively with a finer distortion constraint, followed by an  $\epsilon$ -ball clipping. In [20], project gradient descent (PGD) is introduced, where I-FGSM is modified to incorporate random starts.

***C&W & EAD***. Instead of leveraging the training loss, C&W [7] designed an  $L_2$ -regularized loss function based on the logit layer representation in DNNs for crafting adversarial examples. Handling the box constraint  $x \in [0, 1]^p$  using a change of variables, they used Adam [15] to minimize  $c \cdot f(x, t) + \|x - x_0\|_2^2$ , where  $f(x, t)$  is a loss function depending logit layer values and target class  $t$ . EAD [8] generalizes the attack by minimizing an additional  $L_1$  penalty, and has been shown to generate more robust and transferable adversarial examples [19, 24, 25].

White-box attacks can also be used in black-box settings by taking advantage of *transferability* [18]. Transferability refers to the property that adversarial examples generated using one model are often misclassified by another model [27]. The substitute model approach to black-box attacks takes advantage of this property to generate successful adversarial examples, as we will discuss in the next subsection.

### 2.2 Black-box attacks

In the literature, the *black-box* attack setting has been referred to as the case where an attacker has free access to the input and output of a targeted DNN but is unable to perform back propagation on the network. Proposed approaches have relied on transferability and gradient estimation, and are summarized below.

***Substitute Networks***. Early approaches to black-box attacks made use of the power of free query to train a substitute model, a representative substitute of the targeted DNN [22]. The substitute

DNN can then be attacked using any white-box technique, and the generated adversarial examples are used to attack the target DNN. As the substitute model is trained to be representative of a targeted DNN in terms of its classification rules, adversarial examples of the substitute model are expected to be similar to adversarial examples of the corresponding targeted DNN. This approach, however, relies on the transferability property rather than directly attacking the target DNN, which is imperfect and thus limits the strength of the adversary. Furthermore, training a substitute model is computationally expensive and hardly feasible when attacking large models, such as Inception-v3 [28].

**Zeroth Order Optimization (ZOO).** The zeroth order optimization (ZOO) attack builds on the C&W attack to perform black-box attacks [9], by modifying the loss function such that it only depends on the output of the DNN, and performing optimization with gradient estimates obtained via finite differences. ZOO, however, suffers from requiring a huge number of queries, since a gradient estimate requires 2 queries per each coordinate. Thus, attacking Inception-v3 [28] on the ImageNet dataset requires  $299 \times 299 \times 3 \times 2 = 536,406$  queries per each optimization step. To resolve this issue, stochastic coordinate descent (SCD) is used, which only requires 2 queries per step. Still, convergence of SCD can be slow when the number of coordinates is large, thus reducing the dimensionality of the perturbation and using importance sampling are also crucial. Applying these techniques, unlike the substitute model approach, attacking Inception-v3 becomes computationally tractable. However, as we demonstrate in our experiments, the gradient estimation procedure is still quite query-inefficient, and thus impractical for attacking real-world systems.

In parallel works, [6, 14, 31] have also studied the problem of efficiency and strength of black-box adversarial attacks, but our work remains unique in its goal and approach. [6] focuses on attacking black-box models with only partial access to the query results. Notably, their method takes, on average, about 72x more queries than ours to achieve success against an undefended ImageNet model. [14] and [31] study the efficiency of black-box attacks under the same threat model we consider, however, both rely on *gradient estimation*, rather than *gradient-free optimization*. [14] estimates the gradient of the expected value of the loss under a parameterized search distribution which can be seen as a finite differences estimate on a random gaussian basis. [31] dispenses with ZOO's coordinate-wise estimation with a scaled random full gradient estimator. Though we treat both efforts as parallel work, for the sake of completeness, we provide a comparison between our “gradient-free” approach and the other query-efficient “gradient-estimation” approaches in our results.

### 2.3 Defending against adversarial attacks

**Adversarial Training.** Adversarial training is typically implemented by augmenting the original training dataset with the label-corrected adversarial examples to retrain the network. In [20], a high capacity network is trained against  $L_\infty$ -constrained PGD, I-FGSM with random starts, yielding strong robustness in the  $L_\infty$  ball, but has been shown to be less robust to attacks optimized on other robustness metrics [23, 24, 32]. In [30], training data is augmented

with perturbations transferred from other models, and was demonstrated to have strong robustness to transferred adversarial examples. We demonstrate in our experimental results that its less robust to query-efficient black-box attacks, such as GenAttack.

**Gradient Obfuscation.** It has been identified that many recently proposed defenses provide apparent robustness to strong adversarial attacks by manipulating the gradients to either be nonexistent or incorrect, dependent on test-time randomness, or simply unusable. Specifically, it was found in analyzing the ICLR 2018 non-certified defenses that claim white-box robustness, 7 of 9 relied on this phenomenon [3]. It has also been shown that FGSM based adversarial training learns to succeed by making the gradients point in the wrong direction [30].

One defense which relies upon gradient obfuscation is utilizing non-differentiable input transformations, such as bit-depth reduction, JPEG compression, and total variance minimization [13]. In the *white-box* case, this defense can be successfully attacked with gradient-based techniques by replacing the non-differentiable transformation with the identity function on the backward pass [3]. Though effective, this approach is not applicable in the *black-box* case, since the attacker requires knowledge of the non-differentiable component. We demonstrate in our experimental results that GenAttack, being gradient-free and thus impervious to said gradient manipulation, can naturally handle such procedures in the *black-box* case. Note that many black-box attacks that require gradient estimation including [9, 14, 31] cannot be directly applied when non-differentiable input transformations exist.

## 3 THREAT MODEL

We consider the following attack scenario. The attacker does not have knowledge about the network architecture, parameters, or training data. The attacker is solely capable of querying the target model as a black-box function:

$$f : \mathbb{R}^d \rightarrow [0, 1]^K$$

where  $d$  is the number of input features and  $K$  is the number of classes. The output of function  $f$  is the set of model prediction scores. Note, that the attacker will *not* have access to intermediate values computed in the network hidden layers, including the logits.

The goal of the attacker is to perform a *targeted* attack. Formally speaking, given a benign input example  $\mathbf{x}$  that is correctly classified by the model, the attacker seeks to find a perturbed example  $\mathbf{x}_{adv}$  for which the network will produce the desired target prediction  $t$  chosen by the attacker from the set of labels  $\{1..K\}$ . Additionally, the attacker also seeks to minimize the  $L_p$  distance, in order to maintain the perceptual similarity between  $\mathbf{x}_{orig}$  and  $\mathbf{x}_{adv}$ . i.e.,

$$\arg \max_{c \in \{1..K\}} f(\mathbf{x}_{adv})_c = t \quad \text{such that } \|\mathbf{x} - \mathbf{x}_{adv}\|_p \leq \delta$$

where the distance norm function  $L_p$  is often chosen as  $L_2$  or  $L_\infty$ .

This threat model is equivalent to that of prior work in black-box attacks [9, 22], and is similar to the chosen-plain-text attack (CPA) in cryptography, where an attacker provides the victim with any chosen plain-text message and observes its encryption cipher output.

## 4 GENATTACK ALGORITHM

GenAttack relies on genetic algorithms, which are population-based gradient-free optimization strategies. Genetic algorithms are inspired by the process of natural selection, iteratively evolving a population  $\mathcal{P}$  of candidate solutions towards better solutions. The population in each iteration is called a *generation*. In each generation, the quality of population members is evaluated using a *fitness* function. “Fitter” solutions are more likely to be selected for breeding the next generation. The next generation is generated through a combination of *crossover* and *mutation*. Crossover is the process of taking more than one parent solution and producing a child solution from them; it is analogous to reproduction and biological crossover. Mutation applies a small random perturbation to the population members, according to a small user-defined mutation probability. This is done in order to increase the diversity of population members and provide better exploration of the search space.

Algorithm 1 describes the operation of GenAttack. The algorithm input is the original example  $\mathbf{x}_{orig}$  and the target classification label  $t$  chosen by the attacker. The algorithm computes an adversarial example  $\mathbf{x}_{adv}$  such that the model classifies  $\mathbf{x}_{adv}$  as  $t$  and  $\|\mathbf{x}_{orig} - \mathbf{x}_{adv}\|_\infty \leq \delta_{max}$ . We define the population size to be  $N$ , the “mutation probability” to be  $\rho$ , and the “mutation range” to be  $\alpha$ .

GenAttack initializes a population of examples around the given input example  $\mathbf{x}_{orig}$  by picking random examples from a uniform distributed defined over the sphere centered at the original example  $\mathbf{x}_{orig}$ , whose radius =  $\delta_{max}$ . This is achieved by adding random noise in the range  $(-\delta_{max}, \delta_{max})$  to each dimension of the input vector  $\mathbf{x}_{orig}$ . Then repeatedly, until a successful example is found, each population members’ fitness is evaluated, parents are selected, and crossover & mutation are performed to form the next generation.

**Fitness function:** The subroutine `ComputeFitness` evaluates the fitness, i.e. quality, of each population member. As the fitness function should reflect the optimization objective, a reasonable choice would be to use the output score given to the target class label directly. However, we find it more efficient to also jointly motivate the decrease in the probability of other classes. We also find that the use of log proves to be helpful in avoiding numeric instability issues. Therefore, we chose the following function:

$$\text{ComputeFitness}(\mathbf{x}) = \log f(\mathbf{x})_t - \log \sum_{j=0, j \neq t}^{j=k} f(\mathbf{x})_c$$

**Selection:** Population members at each iteration are ranked according to their fitness value. Members with higher fitness are more likely to be a part of the next generation while members with lower fitness are more likely to be replaced. We compute the probability of selection for each population member by computing the `Softmax` of the fitness values to turn them into a probability distribution. Then, we independently select random parent pairs among the population members according to that distribution. We also find it important to apply the elitism technique [5], where the *elite* member, the one with highest fitness, of the current generation is guaranteed to become a member of the next generation.

---

### Algorithm 1 Targeted Adversarial Attack using GenAttack

---

**Input:** original example  $\mathbf{x}_{orig}$ , target label  $t$ , maximum  $L_\infty$  distance  $\delta_{max}$ , mutation-range  $\alpha$ , mutation probability  $\rho$ , population size  $N$ ,  $\tau$  sampling temperature.

► **Create initial generation.**

**for**  $i = 1, \dots, N$  in population **do**

$\mathcal{P}_i^0 \leftarrow \mathbf{x}_{orig} + \mathcal{U}(-\delta_{max}, \delta_{max})$

**end for**

**for**  $g = 1, 2 \dots G$  generations **do**

**for**  $i = 1, \dots, N$  in population **do**

$F_i^{g-1} = \text{ComputeFitness}(\mathcal{P}_i^{g-1})$

**end for**

► **Find the elite member.**

$\mathbf{x}_{adv} = \mathcal{P}_{\arg \max_j F_j^{g-1}}^{g-1}$

**if**  $\arg \max_c f(\mathbf{x}_{adv})_c == t$  **then**

**Return:**  $\mathbf{x}_{adv}$  ► **Found successful attack**

**end if**

$\mathcal{P}_1^g = \{\mathbf{x}_{adv}\}$

► **Compute Selection probabilities.**

$probs = \text{Softmax}(F^{g-1} / \tau)$

**for**  $i = 2, \dots, N$  in population **do**

Sample  $parent_1$  from  $\mathcal{P}^{g-1}$  according to  $probs$

Sample  $parent_2$  from  $\mathcal{P}^{g-1}$  according to  $probs$

$child = \text{Crossover}(parent_1, parent_2)$

► **Apply mutations and clipping.**

$child_{mut} = child + \text{Bernoulli}(\rho) * \hookrightarrow \mathcal{U}(-\alpha \delta_{max}, \alpha \delta_{max})$

$child_{mut} = \Pi_{\delta_{max}}(child_{mut}, \mathbf{x}_{orig})$

► **Add mutated child to next generation.**

$\mathcal{P}_i^g = \{child_{mut}\}$

**end for**

► **adaptively update  $\alpha, \rho$  parameters**

$\rho, \alpha = \text{UpdateParameters}(\rho, \alpha)$

**end for**

---

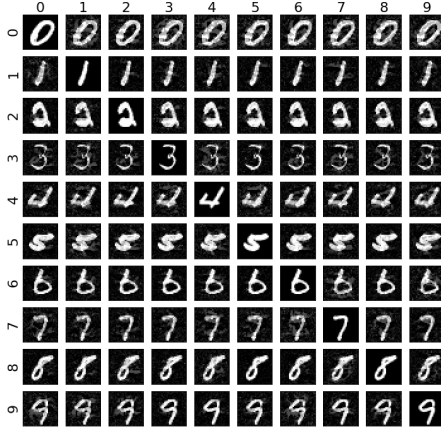
**Crossover operation:** After selection, parents are mated to produce members of the next generation. A child is generated by selecting each feature value from either  $parent_1$  or  $parent_2$  according to the selection probabilities  $(p, 1 - p)$  where  $p$  is defined as

$$p = \frac{\text{fitness}(parent_1)}{\text{fitness}(parent_1) + \text{fitness}(parent_2)}$$

**Mutation operation:** To promote diversity among the population members and exploration of the search space, at the end of each iteration, population members can be subject to mutation, according to probability  $\rho$ . Random noise uniformly sampled in the range  $(-\alpha \delta_{max}, \alpha \delta_{max})$  is applied to individual features of the crossover operation results. Finally, clipping operator  $\Pi_{\delta_{max}}$  is performed to ensure that the pixel values are within the permissible  $L_\infty$  distance away from the benign example  $\mathbf{x}_{orig}$ .

### 4.1 Improving Query Efficiency

In this section, we present a couple of optimizations that we adopt in our GenAttack algorithm, and which contribute significantly to query efficiency.



**Figure 1: MNIST adversarial examples generated by GenAttack. Row label is the true label and column label is the target label.**

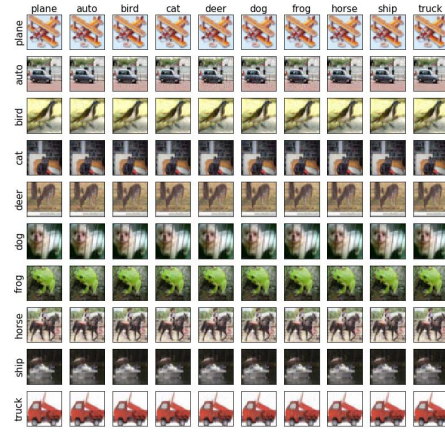
**4.1.1 Dimensionality Reduction:** On one hand, scaling genetic algorithms to explore high-dimensional search spaces (such as ImageNet models) efficiently requires a large population size in each generation. On the other hand, evaluating the fitness of each member implies additional costs in the form of new queries. Therefore, we limit GenAttack to operate using a relatively small (e.g., less than ten) population size. We provide more discussion on the trade-off between population size and number of queries in Section 5.3.

In addition, motivated by the work in [31], we seek to address the challenge of scaling genetic algorithms (in turn GenAttack), by performing dimensionality reduction of the search space and defining adversarial noise in the lower dimensional space. To compute the adversarial example, we apply bilinear resizing (which is deterministic), to scale the noise up to same size as the input. Thus,

$$\mathbf{x}_{orig} \in [0, 1]^d, \quad e_{adv} \in [0, 1]^{d'}, \quad \mathbf{x}_{adv} = S(e_{adv}) + \mathbf{x}_{orig}$$

where  $e_{adv}$  is the learnt adversarial noise,  $S$  is the bilinear resizing operation, and  $d'$  is chosen such that  $d' < d$ . Effectively, this leads to a condensed adversarial noise vector, where one value of  $e_{adv}$  is used to perturb multiple neighbouring pixels of  $\mathbf{x}_{orig}$  to produce  $\mathbf{x}_{adv}$ . We noticed that this approach significantly improves the query efficiency of GenAttack against high dimensional models, such as ImageNet models, while maintaining the attack success rate under the  $L_\infty$  constraint.

**4.1.2 Adaptive Parameter Scaling.** In order to lessen the sensitivity of genetic algorithms to hyperparameter values (e.g. mutation rate, population size, and mutation range), we use an annealing scheme where the algorithm parameters (namely the mutation rate  $\rho$  and mutation range  $\alpha$ ) are decreased gradually if the search algorithm is detected to be stuck for a number of sequential iterations without any further improvement in the objective function. Adaptive scaling alleviates the situation where a very high mutation rate may



**Figure 2: CIFAR-10 adversarial examples generated by GenAttack. Row label is the true label and column label is the target label.**

allow for an initially fast decrease in the objective function value, after which the algorithm may get stuck without achieving any further progress.

We employ exponential decay to update the parameter values

$$\rho = \max(\rho_{\min}, 0.5 \times (0.9)^{\text{num\_plateaus}}) \quad (1)$$

$$\alpha = \max(\alpha_{\min}, 0.4 \times (0.9)^{\text{num\_plateaus}}) \quad (2)$$

where  $\rho_{\min}$  and  $\alpha_{\min}$  are chosen to be 0.1 and 0.15 respectively, and  $\text{num\_plateaus}$  is a counter incremented whenever the algorithm does not improve the fitness of the population's elite member (highest fitness) for 100 consecutive steps.

## 5 RESULTS

We evaluate GenAttack by running experiments attacking state-of-art MNIST, CIFAR-10, and ImageNet image classification models. For each dataset, we use the same models as used in the ZOO work [9]. For MNIST and CIFAR-10, the model accuracies are 99.5% and 80%, respectively. The reader can refer to [7] for more details on the architecture of those models. For ImageNet, we use Inception-v3 [28], which achieves 94.4% top-5 accuracy and 78.8% top-1 accuracy. We compare the effectiveness of GenAttack to ZOO on these models in terms of the attack success rate (ASR), the runtime, and the median number of queries necessary for success. The runtime and query count statistics are computed over successful attacks only. A *single* query means an evaluation of the target model output on a *single* input image. Using the authors' code<sup>2</sup>, we configure ZOO for each dataset based on the implementations the authors used for generating their experimental results [9]. We also evaluate against the state-of-the-art *white-box* C&W attack [7], assuming direct access to the model, to give perspective on attack success rate.

<sup>2</sup><https://github.com/huanzhang12/ZOO-Attack>

In addition, we evaluate the effectiveness of GenAttack against ensemble adversarial training [30], using models released by the authors at the following link<sup>3</sup>. Ensemble adversarial training is considered to be the state-of-art ImageNet defense against black-box attacks, proven to be effective at providing robustness against transferred attacks in hosted competitions [17, 26, 30]. Finally, we evaluate against recently proposed randomized, non-differentiable input transformation defenses [13] to test GenAttack’s performance against gradient obfuscation. We find that GenAttack can handle such defenses as-is due to its gradient-free nature.

**Hyperparameters.** For all of our MNIST and CIFAR-10 experiments, we limit GenAttack to a maximum of 100,000 queries, and fix the hyperparameters to the following values: mutation probability  $\rho = 5e-2$ , population size  $N = 6$ , and step-size  $\alpha = 1.0$ . For all of our ImageNet experiments, as the images are nearly 100x larger than those of CIFAR-10, we use a maximum of 1,000,000 queries and adaptively update the  $\rho$  and  $\alpha$  parameters as discussed earlier in Section 4. In addition, we experimented both with and without dimensionality reduction ( $d' = 96$ ). To match the mean  $L_\infty$  distortion computed over successful examples of ZOO, we set  $\delta_{max} = \{0.3, 0.05, 0.05\}$ , for our MNIST, CIFAR-10, and ImageNet experiments, respectively. As genetic algorithms have various tuning parameters, we conduct parameter sensitivity studies in Section 5.3. To encourage further work, our code has been made available as open source<sup>4</sup>.

## 5.1 Query Comparison

We compare GenAttack and ZOO by number of queries necessary to achieve success, and provide C&W white-box results to put the ASR in perspective. For MNIST, CIFAR-10, and ImageNet, we use 1000, 1000, and 100 randomly selected and correctly classified images from the test sets. For each image, we select a random target label.

**5.1.1 MNIST and CIFAR-10:** Table 1 shows the results of our experiment. The results show that both ZOO and GenAttack can succeed on the MNIST and CIFAR-10 datasets, however GenAttack is 2,126 times and 2,568 times more efficient on each. A randomly selected set of MNIST and CIFAR-10 test images and their associated adversarial examples targeted to each other label are shown in Figure 1 and Figure 2.

**ImageNet:** Table 2 shows the results of our experiment against normally trained (InceptionV3) and ensemble adversarially trained (Ens4AdvInceptionV3) ImageNet models. To demonstrate the effect of dimensionality reduction and adaptive parameter scaling, we denote GenAttack without such tricks as “GA baseline”. On ImageNet, ZOO is not able to succeed consistently in the targeted case, which is significant as it shows that GenAttack is efficient enough to effectively scale to ImageNet. Moreover, GenAttack is roughly 237 times more efficient than ZOO, and 9 times more query efficient than the GA baseline. A random example of results against Inception-v3 test image with its associated adversarial example is shown in Figure 3.

<sup>3</sup>[https://github.com/tensorflow/models/tree/master/research/adv\\_imagenet\\_models](https://github.com/tensorflow/models/tree/master/research/adv_imagenet_models)

<sup>4</sup>[https://github.com/nesl/adversarial\\_genattack](https://github.com/nesl/adversarial_genattack)

	MNIST ( $L_\infty = 0.30$ )		CIFAR-10 ( $L_\infty = 0.05$ )	
	ASR	Queries	ASR	Queries
C&W	100%	–	100%	–
ZOO	98%	2,118,222	93.3%	2,064,798
GenAttack	100%	996	96.5%	804

**Table 1: Attack success rate (ASR) and median number of queries for the C&W (white-box) attack, ZOO, and GenAttack with equivalent  $L_\infty$  distortion. Median of query counts is computed over successful examples. Number of queries is not a concern for white-box attacks.**

	InceptionV3		Ens4AdvInceptionV3	
	ASR	Queries	ASR	Queries
C&W	100%	–	100%	–
ZOO	18%	2,611,456	6%	3,584,623
GA baseline	100%	97,493	93%	163,995
GenAttack	100%	11,081	95%	21,966

**Table 2: Attack success rate (ASR) and median number of queries for the C&W (white-box) attack, ZOO, and GenAttack with equivalent  $L_\infty$  distortion (0.05). Median of query counts is computed over successful examples. GA baseline is GenAttack without the dimensionality reduction and adaptive parameter scaling tricks.**

Attack	Queries Count	$L_2$ -distance	$L_\infty$ -distance
GenAttack	<b>11,081</b>	$2.3 \times 10^{-4}$	<b>0.05</b>
AutoZOOM [31]	13,099	$8.1 \times 10^{-4}$	0.75
Ilyas et al.[14]	14,737	$1.9 \times 10^{-4}$	<b>0.05</b>

**Table 3: Comparison with parallel work against the ImageNet InceptionV3 model in terms of both median of query counts and per-pixel- $L_2$  and  $L_\infty$  distances between the original and adversarial images.**

**5.1.2 Comparison to parallel efforts in query efficient attacks.** While preparing this manuscript, we became aware of parallel efforts that were also developed to address query-efficient adversarial attacks. For sake of completeness, we also present a comparison between our approach and other contributions. Unlike GenAttack, which performs gradient-free optimization, [31] and [14] propose more efficient gradient estimation procedures than ZOO. Table 3 shows a comparison between the results of the three methods. We notice that while the three methods are all significantly more query efficient than the previous state-of-art (ZOO), GenAttack requires 25% fewer queries than the work of [14] under the same  $L_\infty$  distance constraint at the cost of a slight increase in  $L_2$  distance, mainly due to the use of dimensionality reduction. Also, GenAttack requires 15% less queries than [31], even though [31] has higher distortion in both  $L_\infty$  and  $L_2$  distortion. Notably, [31] has additional post-processing steps to reduce the amount of distortion but it significantly costs more queries. Therefore, we reported the number of queries and distortion distances at the first success for all attacks.



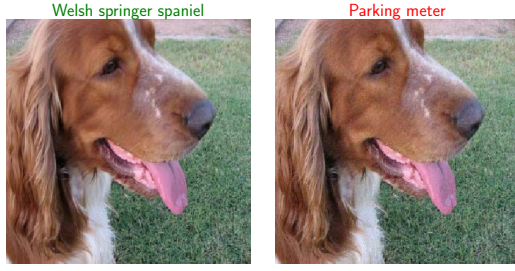


Figure 3: Adversarial example generated by GenAttack against the InceptionV3 model ( $L_\infty = 0.05$ ). Left figure: original, right figure: adversarial example.



Figure 4: Adversarial example generated by GenAttack against the JPEG compression defense ( $L_\infty = 0.15$ ). Left figure: original, right figure: adversarial example.

## 5.2 Attacking Defenses

In the following section, we present how GenAttack succeeds in breaking a set of state-of-the-art defense methods proposed to increase the robustness of models against adversarial attacks.

**5.2.1 Attacking Ensemble Adversarial Training:** Ensemble adversarial training incorporates adversarial inputs generated on other already trained models into the model’s training data in order to increase its robustness to adversarial examples [30]. This has proven to be the most effective approach at providing robustness against transfer-based black-box attacks during the NIPS 2017 Competition. We demonstrate that the defense is much less robust against query-efficient black-box attacks, such as GenAttack.

We performed an experiment to evaluate the effectiveness of GenAttack against the ensemble adversarially trained model released by the authors [30], namely *Ens4AdvInceptionV3*. We use the same 100 randomly sampled test images and targets used in our previous ImageNet experiments. We find that GenAttack is able to achieve 95% success against this strongly defended model, significantly outperforming ZOO. As seen, in Table 2, we compare the success rate and median query count between the ensemble adversarially trained and the vanilla Inception-v3 models. Our comparison shows that these positive results are yielded with only a limited increase in query count. We additionally note that the maximum  $L_\infty$  distortion used for evaluation in the NIPS 2017 competition varied between 4 and 16 (in a 0-255 scale), which when normalized equals 0.02 and 0.06, respectively. Our  $\delta_{max}$  (0.05) falls in this range.

**5.2.2 Attacking Non-Differentiable, Randomized Input Transformations:** Non-differentiable input transformations perform gradient obfuscation, relying upon manipulating the gradients to succeed against gradient-based attackers [3]. In addition, randomized transformations increase the difficulty for the attacker to guarantee success. One can circumvent such approaches by modifying the core defense module performing the gradient obfuscation, however this is clearly not applicable in the black-box case.

In [13], a number of input transformations were explored, including bit-depth reduction, JPEG compression, and total variance minimization. Bit-depth reduction and JPEG compression are non-differentiable, while total variance minimization introduces additional randomization and is quite slow, making it difficult to iterate upon. We demonstrate that GenAttack can succeed against these input transformations in the black-box case, due to its *gradient-free*

	CIFAR-10		ImageNet	
	ASR	Queries	ASR	Queries
Bit depth	93%	2,796	95%	16,301
JPEG	88%	3,541	89%	23,822
TVM	70%	$5,888 \times 32$	—	—

Table 4: Evaluation of GenAttack against non-differentiable and randomized input transformation defenses. We use  $L_\infty = 0.05$  for bit-depth, and  $L_\infty = 0.15$  for JPEG and TVM experiments. For TVM, we compute the expectation of the fitness function by taking  $t = 32$  queries.

and *multi-modal* population-based nature making it impervious to gradient obfuscation. To the best of our knowledge, we are the first to demonstrate a black-box algorithm which can bypass such defenses. Our results are summarized in Table 4.

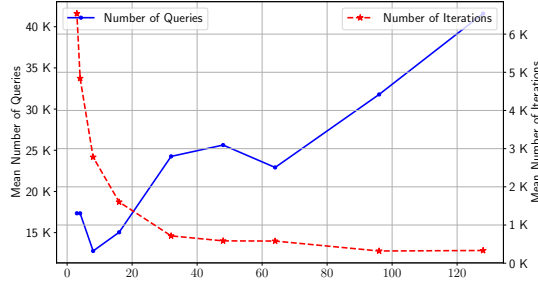
For bit-depth reduction, 3 bits were reduced, while for JPEG compression, the quality level was set to 75, as in [13]. GenAttack is able to achieve high success rate against both non-differentiable transformations, on both the CIFAR-10 and ImageNet datasets. A visual example of our results against JPEG compression is shown in Figure 4.

Total variance minimization (TVM) introduces an additional challenge as it is not only non-differentiable, but it also introduces randomization and is an exceedingly slow procedure. TVM randomly drops many of the pixels (dropout rate of 50%, as in [13]) in the original image and reconstructs the input image from the remaining pixels by solving a denoising optimization problem. Due to randomization, the classifier returns a different score at each run for the same input, confusing the attacker. Succeeding against randomization requires more iterations, but iterating upon the defense is difficult due to the slow speed of TVM processing.

In the setting with randomization, the *ComputeFitness* function can be generalized to be

$$\text{ComputeFitness}(\mathbf{x}) = \mathbb{E}_r [\log f(\mathbf{x}, r)_t - \log \max_{c \neq t} f(\mathbf{x}, r)_c]$$

where  $f(\mathbf{x}, r)$  is the randomization-defended model query function and  $r$  is the noise input to the TVM function, GenAttack can still handle this defense in the black-box case. The expectation is computed by querying the model  $t$  times (we used  $t = 32$ ) for every population member to obtain a robust fitness score at the cost of an increased number of queries. Due to the computational complexity



**Figure 5: Effect of population size selection on both the speed of convergence and the number of queries.**

of applying TVM on each query, we performed the TVM experiment only using the CIFAR-10 dataset and achieved 70% success with  $L_\infty = 0.15$ . Due to the large randomization introduced by TVM, we counted an adversarial example as success only if it is classified as the target label three times in a row. Notably, TVM significantly decreases the model accuracy on clean inputs (e.g. in our CIFAR-10 experiments, from 80% to 40%) unless the model is re-trained with transformed examples [13].

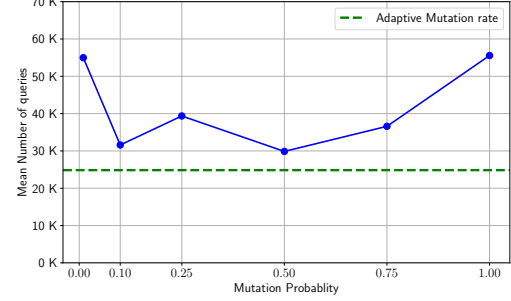
**Comparison to ZOO and C&W:** Due to the non-differentiable nature of the input transformations, the C&W attack, a gradient-based attack, can not succeed without manipulating the non-differentiable component, as discussed in [3]. In the *white-box* case, this method can be applied to yield high success rate, but is not applicable in the more restricted *black-box* case. In the black-box setting, ZOO achieved 8% and 0% against the non-differentiable bit-depth reduction and JPEG compression defenses on ImageNet, again demonstrating its impracticality.

### 5.3 Hyper-parameters values selection

Since genetic algorithms are traditionally sensitive to the choice of hyper-parameter values (e.g. population, mutation rate, etc.), we present a discussion regarding this effect, in the context of query efficiency, which leads to our selection of the hyper-parameter values listed in Section 5.

**Population size:** Large population size allows for increased population diversity and thus improved exploration of the search space within fewer iterations. However, since the evaluation of each population member costs one query, there is a trade-off in the selection of large population size to accelerate the algorithm success (in terms of the number of iterations), and the total number of queries spent. Figure 5 demonstrates this trade-off. On a set of 20 images, we recorded the mean number of queries as well as the number of iterations until success under different choices of population size. From this experiment, we conclude that the relatively small population size of six is a reasonable choice to balance between convergence speed and number of queries.

**Mutation rate:** For the mutation rate  $p$ , we found that the best result can be achieved if we use an adaptive mutation rate which is gradually decayed according to Eq. (1) and (2) in Section 4. As



**Figure 6: Effect of mutation probability on the number of queries required until success.**

shown in Figure 6, this method performs better than fixed valued mutation rates. Effectively, the adaptive mutation rate balances between exploration and exploitation by encouraging exploration initially, and then gradually increasing the exploitation rate as the algorithm approaches convergence near an optimal solution.

## 6 CONCLUSION

GenAttack is a powerful and efficient black-box attack which uses a gradient-free optimization scheme via adopting a population-based approach through genetic algorithms. We evaluated GenAttack by attacking well-trained MNIST, CIFAR-10, and ImageNet models, and found that GenAttack is successful at performing targeted black-box attacks against these models with not only significantly less queries than the previous state-of-the-art, but additionally can achieve a high success rate on ImageNet, which previous approaches are incapable of scaling to. Furthermore, we demonstrate that GenAttack can succeed against ensemble adversarial training, the state-of-the-art ImageNet defense, with only a limited increase in the number of queries. Finally, we showed that GenAttack can succeed against gradient obfuscation, due to its gradient-free nature, namely through evaluating against non-differentiable input transformations, and can even succeed against randomized ones by generalizing the fitness function to compute an expectation over the transformation. To the best of our knowledge, this is the first demonstration of a black-box attack which can succeed against these state-of-the-art defenses. Our results suggest that population-based optimization opens up a promising research direction into effective gradient-free black-box attacks.

## ACKNOWLEDGMENTS

This research was supported in part by the U.S. Army Research Laboratory and the UK Ministry of Defence under Agreement Number W911NF-16-3-0001, the National Science Foundation under awards # CNS-1705135, IIS-1719097, and OAC-1640813, and the NIH Center of Excellence for Mobile Sensor Data-to-Knowledge (MD2K) under award 1-U54EB020404-01. Any findings in this material are those of the author(s) and do not reflect the views of any of the above funding agencies. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.



## REFERENCES

- [1] Moustaafa Alzantot, Bharathan Balaji, and Mani Srivastava. 2017. Did you hear that? adversarial examples against automatic speech recognition. *Machine Deception Workshop, Neural Information Processing Systems (NIPS) 2017* (2017).
- [2] Moustaafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating Natural Language Adversarial Examples. *EMNLP: Conference on Empirical Methods in Natural Language Processing* (2018).
- [3] A. Athalye, N. Carlini, and D. Wagner. 2018. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420* (2018).
- [4] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok. 2017. Synthesizing robust adversarial examples. *arXiv preprint arXiv:1707.07397* (2017).
- [5] Dinabandhu Bhandari, CA Murthy, and Sankar K Pal. 1996. Genetic algorithm with elitist model and its convergence. *International journal of pattern recognition and artificial intelligence* 10, 06 (1996), 731–747.
- [6] Wieland Brendel, Jonas Rauber, and Matthias Bethge. 2018. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *International Conference on Learning Representations (ICLR)* (2018).
- [7] N. Carlini and D. Wagner. 2017. Towards evaluating the robustness of neural networks. *arXiv preprint arXiv:1608.04644* (2017).
- [8] P. Y. Chen, Y. Sharma, H. Zhang, J. Yi, and C. Hsieh. 2017. EAD: Elastic-net attacks to deep neural networks via adversarial examples. *arXiv preprint arXiv:1709.0414* (2017).
- [9] P. Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C. 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*. ACM, 15–26.
- [10] J. Deng, W. Dong, R. Socher, J. Li, K. Li, and L. Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 248–255.
- [11] I. Goodfellow, J. Shlens, and C. Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).
- [12] S. Gu and L. Rigazio. 2014. Towards deep neural network architectures robust to adversarial examples. *arXiv preprint arXiv:1412.5068* (2014).
- [13] C. Guo, M. Rana, and L. van der Maaten. 2017. Countering adversarial images using input transformations. *arXiv preprint arXiv:1711.00117* (2017).
- [14] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin. 2018. Black-box Adversarial Attacks with Limited Queries and Information. *arXiv preprint arXiv:1804.08598* (2018).
- [15] D. Kingma and J. Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
- [16] A. Kurakin, I. Goodfellow, and S. Bengio. 2016. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533* (2016).
- [17] A. Kurakin, I. Goodfellow, S. Bengio, Y. Dong, F. Liao, M. Liang, T. Pang, J. Zhu, X. Hu, C. Xie, J. Wang, Z. Zhang, Z. Ren, A. Yuille, S. Huang, Y. Zhao, Y. Zhao, Z. Han, J. Long, Y. Berdibekov, T. Akiba, S. Tokui, and M. Abe. 2018. Adversarial attacks and defences competition. *arXiv preprint arXiv:1804.00097* (2018).
- [18] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. 2018. Delving into transferable adversarial examples and black-box attacks. *International Conference on Learning Representations (ICLR)* (2018).
- [19] P. H. Lu, P. Y. Chen, K. C. Chen, and C. M. Yu. 2018. On the limitation of MagNet defense against L1-based adversarial examples. *arXiv preprint arXiv:1805.00310* (2018).
- [20] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083* (2017).
- [21] S. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. 2016. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [22] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami. 2017. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*. ACM, 506–519.
- [23] Lukas Schott, Jonas Rauber, Matthias Bethge, and Wieland Brendel. 2019. Towards the first adversarially robust neural network model on MNIST. *International Conference on Learning Representations (ICLR)* (2019).
- [24] Y. Sharma and P. Y. Chen. 2017. Attacking the Madry defense model with L1-based adversarial examples. *arXiv preprint arXiv:1710.10733* (2017).
- [25] Y. Sharma and P. Y. Chen. 2018. Bypassing feature squeezing by increasing adversary strength. *arXiv preprint arXiv:1803.09868* (2018).
- [26] Y. Sharma, T. Le, and M. Alzantot. 2018. CAAD 2018: Generating Transferable Adversarial Examples. *arXiv preprint arXiv:1810.01268* (2018).
- [27] Dong Su, Huan Zhang, Hongge Chen, Jinfeng Yi, Pin-Yu Chen, and Yupeng Gao. 2018. Is Robustness the Cost of Accuracy?—A Comprehensive Study on the Robustness of 18 Deep Image Classification Models. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 631–648.
- [28] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. 2015. Going deeper with convolutions. *CVPR*.
- [29] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, and I. Goodfellow. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199* (2013).
- [30] F. Tramèr, A. Kurakin, N. Papernot, D. Boneh, and P. McDaniel. 2017. Ensemble adversarial training: Attacks and Defenses. *arXiv preprint arXiv:1705.07204* (2017).
- [31] Chun-Chen Tu, Paishun Ting, Pin-Yu Chen, Sijia Liu, Huan Zhang, Jinfeng Yi, Cho-Jui Hsieh, and Shin-Ming Cheng. 2018. AutoZOOM: Autoencoder-based Zeroth Order Optimization Method for Attacking Black-box Neural Networks. *arXiv preprint arXiv:1805.11770* (2018).
- [32] Huan Zhang, Hongge Chen, Zhao Song, Duane Boning, Inderjit S Dhillon, and Cho-Jui Hsieh. 2019. The Limitations of Adversarial Training and the Blind-Spot Attack. *International Conference on Learning Representations (ICLR)* (2019).