SPEAR VS SHIELD: OFFENSIVE AND DEFENSIVE STRATEGIES IN BLACK-BOX ADVERSARIAL ATTACKS

Kwong Yu Chong, Yiyu Lin, Jiahua Wang, Yiliang Yuan, Jingan Zhou
Department of Statistical Science
Duke University
Durham, NC 27708, USA
{kwongyu.chong, yiyu.lin, jiahua.wang, yiliang.yuan, jingan.zhou}@duke.edu

ABSTRACT

This study explores black-box adversarial attacks and their countermeasures in neural networks. Utilizing the PyTorch framework, we implemented attack algorithms and then employed a novel defense mechanism, the post-process adversarial attack on attackers (AAA), to assess system robustness. Our study was conducted on datasets from Tiny ImageNet and Butterfly & Moths Image Classification under constraints including query-limited, partial-information, and label-only settings. Our findings indicate that while the implemented adversarial attack algorithms effectively challenged neural network classifiers, AAA demonstrated limitations in defense, particularly under high-query scenarios and specific attack settings.

1 INTRODUCTION & RELATED WORK

Adversarial attacks, particularly against neural network classifiers, present a significant challenge in the field. They involve creating misleading data that can cause a neural network to malfunction, while still appearing normal to humans. These attacks are broadly categorized into white-box attacks, with full access to model gradients, and black-box attacks, where such information is unavailable. This distinction highlights the realistic and more challenging nature of black-box attacks.

Ilyas et al. (2018) further explored black-box adversarial attacks under three practical constraints, reflecting real-life limitations. These include query-limited scenarios, where the attackers have a finite number of model queries; partial-information scenarios, entailing limited knowledge (usually the probability of the top-k classes); and label-only scenarios, where attackers only access output labels. The key idea in their approach was to estimate the gradient needed for the attack in these constrained settings. Their findings revealed a startling proficiency in breaching model security, achieving success rates above 90%. The results highlight the serious threats posed by such constrained adversarial attacks.

In response to these threats, Chen et al. (2022) proposed a novel defense strategy, termed Adversarial Attack on Attackers (AAA), which effectively countered black-box scored-based query attacks. This method involved post-processing modifications to the output logits, manipulating the loss trend to mislead attackers into incorrect directions. By strategically altering the loss function, AAA successfully confounded attackers, without significantly impacting the model's performance on legitimate inputs. This approach not only preserves the accuracy of clean data but also improves model calibration. The effectiveness of this defense lies in its ability to disrupt the attackers' strategy while maintaining the integrity of the model's outputs.

Our study utilizes the algorithms proposed in these papers to explore both offensive and defensive aspects. We first develop these algorithms implemented via the PyTorch framework. We then apply these strategies to two datasets: Tiny ImageNet and Butterfly & Moths Image Classification. This approach is akin to the ancient Chinese fable of the ultimate spear versus the ultimate shield, testing the strength and effectiveness of each in a balanced confrontation. Our investigation aims to determine which prevails in the realm of neural network security: the spear of sophisticated adversarial attacks or the shield of robust defense mechanisms.

2 Algorithms

2.1 Attack Algorithms

The key idea behind the attack algorithms is to estimate the gradients, and then perform gradient descent in the direction of lowering the original label's logit or perform gradient ascent towards the adversarial label.

2.1.1 NES Gradient Estimation

Algorithm 1 NES Gradient Estimate

```
Input:
    Trained Classifier $\mathbb{P}(y|x)$
    Original Label or Adversarial Target Class Label $y$
    Image $x$
Output:
    Estimated Gradient $g$ of $\mathbb{P}(y|x)$
Parameters:
    Search Variance for Perturbations ($\sigma$)
    Number of Samples for Estimation $n$
    Image Dimension $N$
Algorithm:
$g \leftarrow 0_n$
for i = 1 to $n$ do
    $u_i \leftarrow N(0_N, I_N)$
    $g \leftarrow g + \mathbb{P}(y|x + \sigma \cdot u_i) \cdot u_i$ w.r.t. $y$
    $g \leftarrow g - \mathbb{P}(y|x - \sigma \cdot u_i) \cdot u_i$ w.r.t. $y$
end for
return $\frac{1}{2n\sigma} g$
```

2.1.2 Query-Limited Attack

Algorithm 2 Adversarial Image Generator - Query-Limited

```
Input:
    Trained Classifier $\mathbb{P}(y|x)$
    Original Label $y$
    Initial Image $x$
Output:
    Adversarial Image $x_{adv}$ with $||x_{adv} - x||_\infty \leq \epsilon$
Parameters:
    NES Parameters $(\sigma, n, N)$
    Perturbation Bound $\epsilon$
    Query Limit $L$
    Learning Rate $\eta$
Algorithm:
$x_{adv} \leftarrow x$
for i = 1 to $\frac{L}{2n}$ do
    $g \leftarrow$ NES$(\mathbb{P}(y|x))$
    $x_{adv} \leftarrow x_{adv} - \eta \cdot$ sign$(g)$
    $x_{adv} \leftarrow$ CLIP$(x_{adv}, \max\{x - \epsilon, 0\}, \min\{x + \epsilon, 1\})$
    if $\max_{y_{adv}} \mathbb{P}(y_{adv}|x_{adv}) \neq y$ then
        return $x_{adv}$
    end if
end for
```

2.1.3 Partial-Information Setting

The partial-info and the label-only settings require an input target class y_{adv} and performing gradient ascend towards the target class logit. To ensure consistency, we select the target class to be the class with the second highest probability of occurring. During the gradient estimation, we mask out information not belonging to the top-k classes. The detailed algorithm is shown below.

Algorithm 3 Adversarial Image Generator - Partial-Info and Label-Only

```
Input:
    Trained Classifier $\mathbb{P}(y|x)$
    Initial Image $x$
Output:
    Adversarial Image $x_{adv}$ with $||x_{adv} - x||_\infty \leq \epsilon$
Parameters:
    NES Parameters $(\sigma, n, N)$
    Perturbation Bound $\epsilon$
    Starting Perturbation Bound $\epsilon_0$
    Epsilon Decay Rate $\delta_\epsilon$
    Maximum Learning Rate $\eta_{\max}$
    Minimum Learning Rate $\eta_{\min}$
Algorithm:
$y_{adv} \leftarrow \max_{y \neq y_{\max}} \mathbb{P}(y|x)$
$x_{adv} \leftarrow$ image of the target class $y_{adv}$
$x_{adv} \leftarrow \text{CLIP}(x_{adv}, \max\{x - \epsilon_0, 0\}, \min\{x + \epsilon_0, 1\})$
while $\epsilon_0 > \epsilon$ or $\max_{y_0} \mathbb{P}(y_0|x) \neq y_{adv}$ do
    $g \leftarrow \text{NES}(\mathbb{P}(y_{adv}|x_{adv}))$
    $\eta \leftarrow \eta_{\max}$
    $\hat{x_{adv}} \leftarrow \text{CLIP}(x_{adv} + \eta g, \max\{x - \epsilon_0, 0\}, \min\{x + \epsilon_0, 1\})$
    while $y_{adv} \neq \max_{y_0} \mathbb{P}(y_0|\hat{x_{adv}})$ do
        if $\eta < \eta_{\min}$ then
            $\epsilon_0 \leftarrow \epsilon_0 + \delta_\epsilon$
            $\delta_\epsilon \leftarrow \delta_\epsilon / 2$
            $\hat{x_{adv}} \leftarrow x_{adv}$
            break
        end if
        $\eta \leftarrow \eta / 2$
        $\hat{x_{adv}} \leftarrow \text{CLIP}(x_{adv} + \eta g, \max\{x - \epsilon, 0\}, \min\{x + \epsilon, 1\})$
    end while
    $x_{adv} \leftarrow \hat{x_{adv}}$
    $\epsilon_0 \leftarrow \epsilon_0 - \delta_\epsilon$
end while
return $x_{adv}$
```

2.1.4 Label-Only Setting

In the label-only setting of black-box adversarial attacks, we employ an alternative method to evaluate the success of an adversarial example, bypassing the need for output scores. We introduce the concept of a discretized score, defined as $R(x^{(t)}) = k - \text{rank}(y_{adv}|x^{(t)})$. Here, $\text{rank}(y_{adv}|x^{(t)})$ represents the minimum k for which the adversarial target class y_{adv} appears in the top-k classifications, with a score of 0 if it is not in the top-k.

To approximate the softmax probability, we assess the adversarial image's robustness against random perturbations within a defined radius μ. This robustness is quantified using the discretized score: $S(x^{(t)}) = \mathbb{E}_{\delta \sim U[-\mu,\mu]}[R(x^{(t)} + \delta)]$. The Monte Carlo approximation method is utilized for practical estimation: $\hat{S}(x^t) = \frac{1}{n} \sum_{i=1}^{n} R(x^{(t)} + \mu\delta_i)$. This estimated score $\hat{S}(x)$ acts as a surrogate for the output probability $\mathbb{P}(y_{adv}|x)$, enabling us to apply partial-information techniques and NES gradient estimation to advance the attack.

2.2 Defense Mechanisms

Adversarial Attack on Attackers (AAA), proposed by Chen et al. (2022), is a post-processing defense against black-box score-based query attacks (SQAs). In SQAs, attackers perform greedy updates on adversarial examples (AEs) following attack directions (loss change) indicated by models' output scores (logits or probabilities), repeatedly updating the AEs to minimize the loss (maximizing the probability of target class) while maintaining a certain distance from the original image. AAA directly counters the greedy update of AEs by performing an optimization to perturb the model output scores and mislead attackers while maintaining output confidence. In other words, it post-processes output scores such that gradients indicated by strategies like NES are less reliable. At the same time, it ensures that the confidence of the modified logits remains close to the original.

The mathematical objective of AAA is defined as follows. Suppose that a SQA on a neural network that outputs logits $f(x)$ aims to generate AE x by minimizing the margin between logits:

$$L(f(x), y) = f_y(x) - \max_{k \neq y} f_k(x)$$

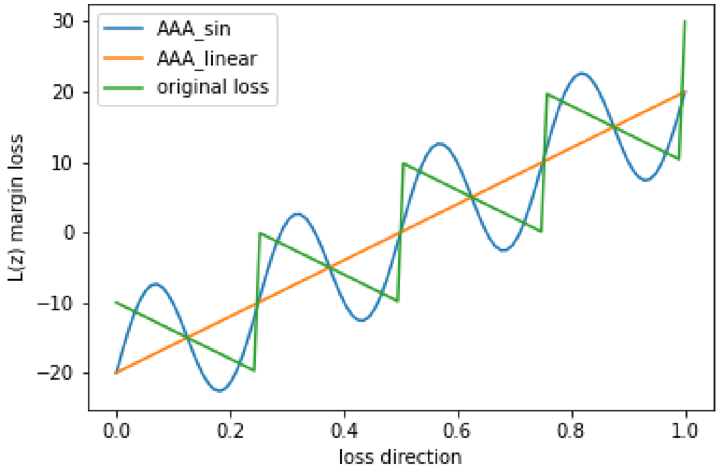
Since true class y is often unknown to the defender, predicted class $\hat{y}$ is used instead, which gives $L(z, \hat{y}) = L_u(z)$. AAA perturbs the loss to a target loss that misleads the attackers by optimizing:

$$\min_z ||L_u(z) - l_{trg}||_1 + \beta ||\sigma(z) - p_{trg}||_1$$

where $z = f(x)$ is the logits, $\sigma(z) = \max(\text{softmax}(z))$, $p_{trg} = \sigma(z_{org}/T)$ and z_{org} is the raw logits output from model. The first term intends to perturb the loss to mislead the attacker, whereas the second term aims to preserve the model's confidence in its prediction, and β controls the trade-off between the two goals.

The target loss l_{trg} can be designed in many ways, as long as they mislead the attackers in the wrong direction. Chen et al. (2022) proposed linear and sine perturbation that gives the opposite direction to the true loss direction locally, but maintains the true direction globally. They are illustrated in the Figure below.

Figure 1: Visual illustration of AAA



We adopted the linear loss target l_{org} which is given below:

$$l_{atr} = (floor(l_{org}/\tau + 1/2)) * \tau$$
$$l_{trg} = l_{org} - \alpha \times \tau \sin(\pi(1 - 2(l_{org} - l_{atr})/\tau))$$

The idea can be generalized to other loss targets: AAA takes the output logits from the model, computes a locally misleading loss target, and finds a new logits output that optimizes the objective function defined above. The algorithm can be summarized as follows.

Algorithm 4 Adversarial Attack on Attackers (AAA)

```
Require: logits $z_{org}$, loss target generator $l_{trg}$ , Temperature $T$
1: Obtain original loss $l_{org} = L_u(z_{org})$
2: Obtain target loss $l_{trg} = l_{trg}(l_{org})$
3: Obtain target confidence $p_{trg} = \sigma(z_{org}/T)$
4: Initialize $z = z_{org}$
5: Optimize $z^* = \min_z ||L_u(z) - l_{trg}||_1 + \beta||\sigma(z) - p_{trg}||_1$ for $\kappa$ iterations.
6: return $z^*$
```

3 Experiment

3.1 Dataset

In our study, we utilize two datasets available on Kaggle: the Butterfly & Moths Image Classification 100 species dataset (referred to as Butterfly) and a subset of 20 classes from Tiny ImageNet (referred to as ImageNet). Due to computational constraints, we have selected a subset of 20 classes from the ImageNet dataset. The specific details regarding the number of training and testing images for each dataset are presented in Table 1.

Dataset	Number of Training Images	Number of Testing Images	Number of Classes
Butterfly	12,594	500	100
ImageNet	8,000	2,000	20

Table 1: Dataset Information

3.2 Training the Classifiers

Model Architecture. For both datasets, we employed a fine-tuning approach on pre-trained models. For the Butterfly dataset, we utilized a DenseNet121 architecture, and for the ImageNet dataset, an InceptionV3 model. Both models were initialized with pre-trained weights and were further fine-tuned to our specific datasets. This fine-tuning involved adding three additional fully connected layers (from 1000 to 32, then 32 to 64, and finally 64 to the number of classes) to the original architectures, adapting them to the unique requirements of our datasets.

Training Process. Both models were trained over 5 epochs with a batch size of 100, using a learning rate of 0.0001. The AdamW optimizer was used for the Butterfly dataset and the Adam optimizer with an additional weight decay of 1e-4 for the ImageNet dataset. A ReduceLROnPlateau scheduler was employed for ImageNet to optimize the learning rate.

Performance. The DenseNet121 model achieved a testing accuracy of 96.00% on the Butterfly dataset, and the InceptionV3 model reached 83.55% accuracy on the ImageNet dataset. These results demonstrate the effectiveness of the chosen models and training parameters.

The trained classifiers serve as the foundation for the subsequent stages of our study, involving adversarial attacks and defense mechanisms evaluation.

3.3 Correctness Subset

Our research aims to investigate the effectiveness of black-box adversarial attacks in misleading a model to incorrectly classify previously correct predictions. To this end, we focused exclusively on a subset of test set images where the model initially made accurate classifications. In our study, this subset comprised 482 out of 500 images from the Butterfly dataset, and 1928 images from the ImageNet dataset. For a balanced comparison, we randomly selected 500 images from the ImageNet subset as the target for our adversarial attacks.

The exception is the label-only setting, where millions of queries are needed to perform a successful attack. Thus, we evaluate only 2 images for the Butterfly dataset. The defense mechanism is also not attempted as AAA does not apply to decision-based query attacks.

3.4 Hyperparameters for Attack

General	
Query Limit (for Query Limited and Partial Info Settings)	30,000
Searching Variance for NES (σ)	0.001
Half of the Size of each NES Population (n)	50
l_∞ Distance to the Original Image (ϵ)	0.1
Query Limited Attack	
Learning Rate (η)	0.01
Partial Info & Label Only Attack	
Initial Distance from Source Image (ϵ_0)	0.5
Rate at which to Decay ϵ (δ_ϵ)	0.01
Maximum Learning Rate (η_{max})	0.02
Minimum Learning Rate (η_{min})	0.01
Number of Accessible Classes (k)	1
Label Only Attack	
Query Limit (for Label Only Setting, Butterfly Dataset)	3,000,000
l_∞ Radius of Sampling Ball (μ)	0.001
Number of Samples for Proxy Score (m)	50

Table 2: Parameter Settings

3.5 Hyperparameters for Defense

AAA	
Loss Target Function (l_{trg})	Linear
Temperature parameter (T)	1
Period Parameter (τ)	10
Subtractor Parameter (α)	1
Trade-off Parameter (β)	5
Number of Iterations (κ)	100

Table 3: AAA Parameter Settings

3.6 Results

Without defense					With AAA			
Setting	Sample	Successful Attack	Accuracy	Median	Sample	Successful Attack	Accuracy	Median
QL	482	472	0.979	2000	50	29	0.58	12658
PI	482	471	0.977	13235	50	2	0.04	30000
LO	2	0	0	30000	-	-	-	-

Table 4: Results for the Butterfly Dataset without AAA

Without defense					With AAA			
Setting	Sample	Successful Attack	Accuracy	Median	Sample	Successful Attack	Accuracy	Median
QL	500	499	0.998	500	50	29	0.58	12658
PIA	500	320	0.64	14819	10	0	0	30000

Table 5: Results for the Imgnet Dataset without AAA

From the results, attacks under Query-Limited (QL) settings are extremely successful for both datasets, achieving an accuracy of 97.9% and 99.8% for Butterfly and ImageNet respectively. Attacks under Partial-Information-Attack (PIA) are extremely successful for the Butterfly dataset

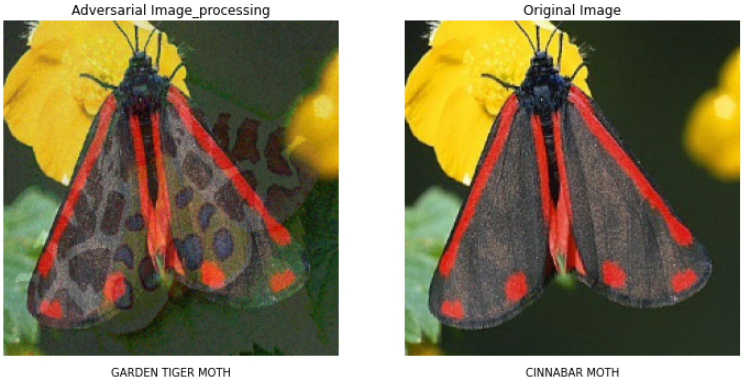


Figure 2: Sample Images from a successful PI Attack

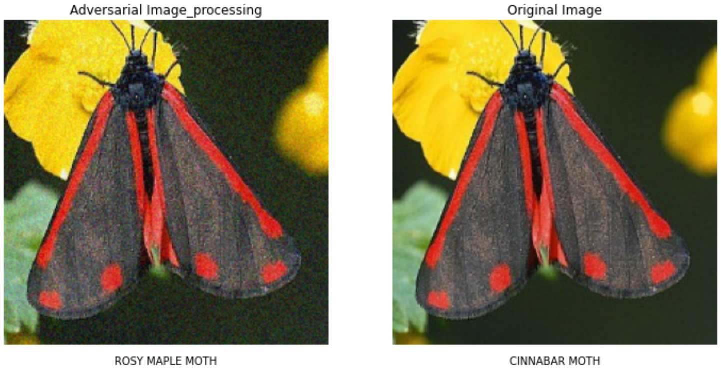


Figure 3: Sample Images from a successful QL Attack

(97.7% accuracy), but mediocre for ImageNet (64% accuracy). Attacks under Label-Only (LO) settings are extremely difficult under realistic settings with limited queries, as the combined number of queries required for estimation of probabilities and gradients through the discretized score and NES grows exponentially. For each gradient estimation, the LO attack requires $2 \times n$ probability outputs, and for each probability output, it requires m queries, leading to $2 \times m \times n = 2 \times 50 \times 50 = 5000$ queries per gradient update. Under our settings of 30000 query limits, LO attacks can only perform 6 updates, which barely does the job. This prompts us to give up experiments on LO altogether. Attacks under LO settings on the ImageNet dataset require median queries of 2.7×10^6, as given in Ilyas et al. (2018).

As for attacks with AAA defense, the experiments are also difficult, since each query requires additional $\kappa = 100$ iterations for the AAA optimization task. However, looking at the 50 attacks under QL and PI for the Butterfly dataset, attack accuracy reduces greatly to 58% and 4% respectively. Moreover, the median queries increase greatly from 2000 to 12658 and 13235 to 30000 (upper limit) respectively for QL and PI. This suggests that the perturbation on logits from AAA can successfully increase the queries required for the attackers, and at the same time mislead their attack directions. As for attacks on the Imagenet dataset with AAA defense, attack accuracy similarly decreases to 58% and 0% under QL and PI settings. The median queries similarly increase greatly. All attacks on PI with AAA failed due to exceeded query limits.

4 Discussion

This section aims to elucidate the underlying principles of the attack and defense methodologies discussed in this paper, extending the idea to explain the numerical results.

At their core, the three adversarial attack algorithms share a common approach. They repeatedly query the model, using the outputs received to approximate the gradient $\nabla_x P(y_{trg}|x)$. This gradient estimation is used to guide the modification of the adversarial example (AE) to either increase or decrease the probability of a designated target class $P(y_{trg}|x)$.

For the Query-Limited (QL) attack, the process involves an approximation of the gradient of the original class $\nabla_x P(y_{org}|x)$ using Natural Evolution Strategies (NES). The image is then adjusted by subtracting this gradient. This action is intended to reduce the probability predicted by the model for the original class $P(y_{org}|x)$. Subsequently, the modified image is subjected to ℓ_∞ projection onto the proximity of the original image. As a result of the above iterative process, the produced AE effectively lowers the probability $P(y_{org}|x)$ while maintaining a resemblance to the original image, leading the model to misclassify.

The strategy for the Partial-Information (PI) setting diverges slightly from the previous approach. It starts with selecting a specific target class and an associated image from that class. The subsequent steps involve an iterative process where this image is gradually adjusted to bring it closer to the original image. During this process, the gradients of the target class $\nabla_x P(y_{trg}|x)$ are estimated using Natural Evolution Strategies (NES) and incorporated into the adversarial example (AE) through the addition of the gradients. Eventually, we obtain an AE that maximizes the model probability of incorrect target class y_{trg} that is at the same time close to the original image. The rationale behind selecting a target class in the PI setting is due to the constraints of our problem: we have access to only the top-k probabilities. To continuously obtain gradient information under these conditions, it is necessary to choose a target class and alter the image to increasingly resemble this class. This approach ensures that the model's output remains relevant to our target, allowing us to consistently retrieve the necessary gradients.

The Label-Only Setting shares similarities with the Partial-Information (PI) approach, but it introduces a distinct twist. Like PI, it involves choosing a target class and iteratively modifying an image from that class. The key difference lies in estimating the unknown probability $P(y_{trg}|x)$ from the output ranks of the model. This is achieved by employing a discretized score based on the output rank. This is a necessary adaptation due to the unique constraints of the Label-Only context, where only ranked top-k class labels are available without probability scores. While effective, this method requires a significantly higher number of queries to the model, increasing the overall computational demand.

The defense algorithm AAA, with its unique optimization goal, primarily influences the top two logits in a model. This is due to the use of the predicted class $\hat{y}$ (the one with the highest original logit) in calculating margin loss, which essentially measures the gap between the largest and second-largest logits. The algorithm's optimization process intentionally adjusts these top two logits, altering their relative distance according to a predetermined pattern, such as linear or sinusoidal.

Following the intuitions, it is not hard to see why the effectiveness of AAA in our tests was good. Firstly, most PIA and LO attacks require a lot of queries, which easily exceed limits, especially with the perturbation from AAA. In such scenarios with perturbations, gradient estimates from Natural Evolution Strategies (NES) suffer from higher variance, which leads to slower convergence of attacks. As for the Query-Limited (QL) setting, as the gradients are constantly reduced from the original image, the originally correct class eventually falls out of the top two predictions. Consequently, its logit is no longer affected by the perturbations of AAA, and hence the defense against QL is slightly worse. In PIA settings, the target adversarial class is maintained at the top-k classes, whose logits are constantly perturbed by AAA, and hence AAA is particularly successful in defending against PIA attacks. As for the Label-Only (LO) setting, attackers deploy decision-based query attacks that focus solely on the rank of output labels, circumventing the logit-based defenses of AAA. This limitation is consistent with observations made in the study by Chen et al. (2022), which highlighted that AAA is not equipped to counter thousands of queries in decision-based query attacks. Still, attacks under LO failed as they easily exceeded query limits.

However, we note that there are too many hyper-parameters in AAA to tune. Find ng the right hyper-parameters that can effectively defend attackers for a specific dataset is challenging, leading to sub-optimal defense performance.

REFERENCES

S. Chen, Z. Huang, Q. Tao, Y. Wu, C. Xie, and X. Huang. Adversarial attack on attackers: Post-process to mitigate black-box score-based query attacks. 2022.

A. Ilyas, L. Engstrom, and A. Madry. Black-box adversarial attacks with limited queries and information. 2018.

WORK CONTRIBUTION

All authors share equal contributions to this paper.

ACKNOWLEDGEMENT

We note a few errors in the original paper by Ilyas et al. (2018). The three algorithms presented in this paper are the modified and corrected versions of those algorithms.