Black-Box Sparse Adversarial Attack via Multi-Objective Optimisation

Phoenix Neale Williams, Ke Li
Department of Computer Science, University of Exeter
Stocker Rd, Exeter, EX4 4PY
{pw384, k.li}@exeter.ac.uk

Abstract

Deep neural networks (DNNs) are susceptible to adversarial images, raising concerns about their reliability in safety-critical tasks. Sparse adversarial attacks, which limit the number of modified pixels, have shown to be highly effective in causing DNNs to misclassify. However, existing methods often struggle to simultaneously minimize the number of modified pixels and the size of the modifications, often requiring a large number of queries and assuming unrestricted access to the targeted DNN. In contrast, other methods that limit the number of modified pixels often permit unbounded modifications, making them easily detectable. To address these limitations, we propose a novel multi-objective sparse attack algorithm that efficiently minimizes the number of modified pixels and their size during the attack process. Our algorithm draws inspiration from evolutionary computation and incorporates a mechanism for prioritizing objectives that aligns with an attacker's goals. Our approach outperforms existing sparse attacks on CIFAR-10 and ImageNet trained DNN classifiers while requiring only a small query budget, attaining competitive attack success rates while perturbing fewer pixels. Overall, our proposed attack algorithm provides a solution to the limitations of current sparse attack methods by jointly minimizing the number of modified pixels and their size. Our results demonstrate the effectiveness of our approach in restricted scenarios, highlighting its potential to enhance DNN security.

1. Introduction

Although deep neural networks (DNNs) have made impressive strides in computer vision tasks [17, 21, 22, 24, 28, 37, 38, 48], recent work has shown that small optimized perturbations to input images can cause DNNs to misclassify [1, 18, 26, 31, 34, 42]. As adversarial images have been found to exist in the physical world [26, 27, 43], particular concern has been expressed on their impact on security-critical applications [1]. To address this issue, previous works have emphasized the importance of generating strong adversarial images [3]. As a result, significant effort has been devoted to developing effective attack methods that can construct perturbations capable of causing DNN classifiers to misclassify images while preserving their semantic content.

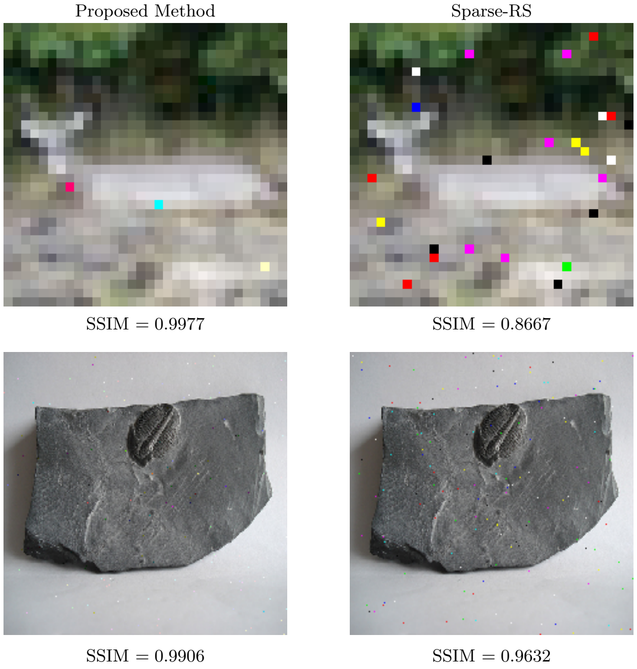

Figure 1. This illustration shows adversarial images generated by two different algorithms, the proposed method and Sparse-RS [10], both attacking an adversarial trained CIFAR-10 [19] (top) and ImageNet [35] (bottom) DNN classifiers. While both images are adversarial, the perturbation generated by the Sparse-RS algorithm visibly distorts the image, whereas the proposed method's adversarial image remains more similar to the original. This similarity is demonstrated by calculating the structural similarity (SSIM) between the adversarial images and the original. The effectiveness of the Sparse-RS algorithm is therefore questionable due to the significant distortion it causes.

Most adversarial attack methods in the literature formulate the attack as an optimization problem where a loss function is minimized to achieve the desired misclassification of an image. While many attack methods in the literature constrain the adversarial perturbation by its l_2 or l_∞ norm [2, 4, 6–8, 18, 23, 26, 29, 36, 42, 45, 46] and allow all pixels of an image to be perturbed, there is also a need to develop sparse attack methods that constrain adversarial perturbations by their l_0 norm [10, 11, 15, 16, 30, 41, 44, 47]. Such adversarial images have been found to also exist in the physical world and have shown to be as effective as the more traditional l_2 or l_∞-constrained adversarial images.

Numerous sparse attack methods have been proposed to address both the white-box [11, 15, 16, 44, 47] and black-box scenarios [10, 30, 41]. In the white-box scenario, the attacker has full access to a DNN's information, while the black-box scenario assumes the attacker only has access to the outputted class probabilities. In this work, we focus on the black-box scenario.

While existing attack methods have shown success in generating adversarial images, they often struggle to handle the trade-off between optimizing the loss function and minimizing the perturbations l_p norms, where $p = 0, 1, 2$ or ∞. Several methods only constrain the number of perturbed pixels [10,11,41], allowing the size of the perturbation to be unbounded. Despite their efficiency, the unbounded nature of the generated perturbations result in obvious distortions of the original image, as shown in Fig. 1. On the other hand, other methods allow the l_0 norm to be minimized along with the loss function [15, 16, 47], while constraining the perturbation by its l_2 or l_∞ norm. These methods either generate an adversarial perturbation and then reduce its l_0 norm [47] or add an l_0 norm penalty term to the optimized loss function [15, 16]. Despite their good performance, these methods only address the white-box scenario and assume access to a large number of DNN queries, limiting their applicability to query-limited scenarios [23]. Therefore, by not properly handling this trade-off, existing methods are limited in their applicability and effectiveness in real-world scenarios.

Within the evolutionary computation field a classic approach to handling conflicting objectives is the use of a domination relation [13] which characterises the trade-off between objectives and is used to compare solutions within a population-based evolutionary algorithm. While the original approach assigns equal weight to each objective, the domination mechanism can be adapted to reflect the attacker's preferences. In this work, our goal is to generate sparse adversarial perturbations with low l_0 and l_2 norms in an efficient manner. Our contributions can be summarised as follows:

- To address the challenge of generating sparse adversarial perturbations, we formulate the problem as a bi-objective optimization problem. By constraining the perturbation to a set of discrete values, we show that minimizing of the l_2 norm also minimizes the l_0 norm.
- We propose a new dominance relation to compare solutions that gives first priority to minimizing the loss function and then to minimizing its l_2 norm.
- To generate adversarial perturbations, we propose a population-based heuristic attack method that utilizes two distributions to generate new solutions. These distributions mimic the crossover and mutation operators commonly used in evolutionary computation. By sampling from these distributions, our approach explores the search space in an efficient and effective manner.
- To evaluate the effectiveness of our approach, we conduct attacks on DNN classifiers trained on the CIFAR-10 and ImageNet datasets, using a low-query budget and considering both targeted and non-targeted attack scenarios. Our empirical results demonstrate that our proposed method outperforms state-of-the-art white-box sparse attacks, as well as the black-box Sparse-RS attack method [10], in terms of success rate and number of perturbed pixels.

2. Related Works

Many works in the literature have proposed attack algorithms that aim to generate l_2 or l_∞ constrained adversarial perturbations. In the white-box scenario, where an attacker has complete access to the targeted DNNs information, many works [6, 7, 11, 18, 26, 29, 36, 42] utilise the gradient information of its loss function within an optimization algorithm to generate adversarial perturbations. One the other hand, in the black-box scenario, where access to the DNN is limited to its outputted probabilities, several attack methods approximate the gradient of the loss function [4, 8, 23, 45, 46] and use it within gradient-based optimization methods. Alternatively, heuristic methods that do not rely on gradient information have also been proposed for the black-box scenario [1, 2].

Sparse adversarial attack, in contrast, aim to modify the smallest number of pixels possible to generate adversarial images [11]. In the white-box scenario, Croce et al. [11] propose a sparse attack method that extends the PGD algorithm of Madry et al. [29] to the l_0 ball. They also utilize a heuristic search method called CornerSearch that searches over all pixels and selects a subset to perturb. The SAPF algorithm of Fan et al. [16], formulates the sparse adversarial attack as a mixed integer programming task and applies a cardinality constraint to control the sparsity of the perturbation. The l_p-Box ADMM [5] algorithm is then utilized for its optimization. Dong et al. [15] recently proposed the GreedyFool algorithm, which generates perturbations using a greedy search method. The method first selects perturbations with large gradients, then removes those that do not impact the desired misclassifiction. The Homotopy attack algorithm of Zhu et al. [47] does not control the sparsity of the perturbation. Instead, they encourage the algorithm to generate increasingly sparse perturbations by adding a weighted l_0 norm penalty term to the optimized loss function. The evolutionary homotopy algorithm is used to optimize the weight of the penalty term while optimizing the value of the perturbation.

While most sparse attacks aim to minimize the l_0 norm of the perturbation, the Sparse-RS attack algorithm introduced by Croce et al. [10] prioritizes query efficiency by allowing the size of the perturbations to be unbounded while constraining the number of modified pixels. To generate an adversarial image, the authors iteratively sample pixel locations from a designed distribution, where the perturbation value of each pixel is uniformly sampled from the corners of a color cube $[-1, 1]$. Given the lack of work in this area, Croce et al. [10] proposed black-box variants of the PGD$_0$ and JSMA algorithms [11, 32] for comparison. The One-Pixel attack method proposed by Su et al. [41] utilizes the differential evolution algorithm [40] to optimize the location and value of a single modified pixel.

Recent works have shown impressive performance gains in sparse adversarial attacks. However, existing methods tend to prioritize either query efficiency by allowing unbounded modifications or l_0 norm minimization by constraining the l_2 or l_∞ norms and making use of a large query budget. Thus, effectively handling the trade-off between these norms is a critical direction to enhance the applicability of sparse attack in realistic scenarios, especially in black-box query-limited settings [23].

3. Proposed Method

In this section, we start with an introduction of our problem formulation. Then, we delineate the implementation of our proposed method step by step.

3.1. Problem Formulation

Let $f : \mathcal{X} \subseteq [0, 1]^{h\times w\times 3} \to \mathbb{R}^K$ be a trained DNN image classifier that takes a benign RGB image $\mathbf{x} \in \mathcal{X}$ of height h and width w as its input and outputs a label $y = \underset{i\in\{1,\cdots,K\}}{\operatorname{argmax}} f_i(\mathbf{x}+\vec{\delta})$, where K is the number of class labels. A non-targeted attack aims to search for a perturbation $\vec{\delta} \in \mathbb{R}^{h\times w\times 3}$ for $\mathbf{x}$ such that:

$$\underset{i\in\{1,\cdots,K\}}{\operatorname{argmax}} f_i(\mathbf{x} + \vec{\delta}) \neq y, \tag{1}$$

where y is the ground truth class label for $\mathbf{x}$. When considering the sparse scenario, the number of perturbed pixels should be small enough to ensure the semantic content of the image unchanged. This problem is thus cast as:

$$\begin{aligned} &\min_{\delta} \mathcal{L}(f; \mathbf{x} + \vec{\delta}, y_q) \\ &\text{s.t. } ||\vec{\delta}||_0 \leq \epsilon, \;\; 0 \leq \mathbf{x} + \vec{\delta} \leq 1 \end{aligned} \tag{2}$$

where $y_q = \underset{q\neq y}{\operatorname{argmax}} f_i(\mathbf{x})$ and the minimization of the loss function $\mathcal{L}$, such as the cross entropy function, leads to the desired adversarial image. Alternatively, an attacker may want the DNN to misclassify to a particular class y_t such that

$$\underset{i=\{1,\cdots,K\}}{\operatorname{argmax}} f_i(\mathbf{x} + \vec{\delta}) = y_t, \tag{3}$$

namely a targeted attack. The problem defined in equation (2) can be adapted to this scenario by minimizing the loss of the classifier with the particular class,

$$\begin{aligned} &\min_{\delta} \mathcal{L}(f; \mathbf{x} + \vec{\delta}, y_t) \\ &\text{s.t. } ||\vec{\delta}||_0 \leq \epsilon, \;\; 0 \leq \mathbf{x} + \vec{\delta} \leq 1. \end{aligned} \tag{4}$$

Existing algorithms in the literature typically focus on solving equation (2) and equation (4) by either fixing the number of perturbed pixel to a specific value ϵ and allowing unbounded modifications [10, 41] or constraining its l_2 or l_∞ norms while attempting to minimize its l_0 norm [11, 15, 16, 47]. In this work, our objective is to efficiently generate sparse adversarial images with low l_0 and l_2 norms by treating the problem as a multi-objective task. Previous work [15] has demonstrated the difficulty of minimizing the l_0 norm of an adversarial perturbation by greedily removing perturbed pixels. Therefore, we argue that an improved method should jointly search for perturbations with small l_0 and l_2 norms whilst minimizing the loss function $\mathcal{L}(\mathbf{x} + \vec{\delta})$. Our aim is to generate a $\vec{\delta}$ that solves the following optimization problem:

$$\begin{aligned} &\min_{\vec{\delta}} F(\mathbf{x} + \vec{\delta}) \\ &\text{s.t. } ||\vec{\delta}||_0 \leq \epsilon, \;\; 0 \leq \mathbf{x} + \vec{\delta} \leq 1, \end{aligned} \tag{5}$$

where $F(\mathbf{x} + \vec{\delta}) = (\mathcal{L}(\mathbf{x} + \vec{\delta}),\ ||\vec{\delta}||_2, ||\vec{\delta}||_0)^\top$ is the objective vector. Here, $\mathcal{L}(\cdot)$ is defined by (4) for targeted attacks and (2) for non-targeted attacks.

Croce et al. [10] demonstrated that perturbation values of $\{-1, 1\}$ were effective under the l_0 norm constraint. To accommodate 0-valued perturbations that reduce the l_2 norm of the perturbation, we define the space of perturbation values as the set $\{-1, 1, 0\}$. This discrete set of values also ensures that the l_0 norm of the perturbation approaches zero as the l_2 norm approaches zero, allowing us to represent the problem as a bi-objective task. Specifically, we define the objective vector as $F(\mathbf{x} + \vec{\delta}) = (\mathcal{L}(\mathbf{x} + \vec{\delta}),\ ||\vec{\delta}||_2)^\top$, where $\vec{\delta} \in \{-1, 1, 0\}$.

3.2. Multi-Objective Optimization

Evolutionary algorithms are optimisation methods inspired by biological evolution. They iterate through initialization, variation, evaluation, selection, and termination steps. By maintaining a population of candidate solutions and applying the operators of crossover and mutation, evolutionary algorithms can explore the search space and locate promising areas, leading to high-quality solutions. Compared to random search methods, they leverage information from previous iterations to bias the search towards promising areas. For multi-objective optimization, evolutionary algorithms maintain a set of non-dominated solutions, known as the Pareto front. This set is determined by the domination mechanism based on objective function values. In this section, we describe the heuristic method used to solve the bi-objective optimization problem in equation (5). We provide the pseudo-code of the proposed method in Algorithm 4 within the appendix.

Initialization. The attack method initializes by setting the number of perturbed pixels to k and constructing a set P of s solutions with uniformly sampled pixel locations from the set $\{1, \cdots, h \cdot w\}$. The initial perturbation values for each color channel are generated by random sampling from the set $\{-1, 1, 0\}$, where the probability of sampling 0 is defined by pr_0.

Crossover. The goal of a crossover operator is to locate promising areas by generating solutions that inherit beneficial traits from their parents. To achieve this, we exchange a random subset of pixel locations and corresponding perturbation values between two solutions. Specifically, given a pair of solutions P_a and P_b with sets of pixel locations M_a, M_b and perturbation values Δ_a, Δ_b, the crossover operator produces two new solutions that combine the pixel locations and perturbation values from their parents.

To perform the crossover, for each solution $r \in \{a, b\}$ and opposing solution $e \in \{b, a\}$, we define $U = M_e \setminus (M_r \cap M_e)$ as the set of disjoint pixel locations between M_r and M_e. If $|U| > 0$, we randomly select $A \subset M_r$ and $B \subset U$ such that $|A| = |B| = \min\{p_c \cdot k, |U|\}$, where k is the maximum number of modified pixels and p_c is a user-defined parameter that specifies the largest percentage of pixels that can be exchanged between the solutions. We then generate the set $M'_r = (M_r \setminus A) \cup B$ of updated pixel locations, and the corresponding perturbation values $\Delta'_r \leftarrow (\Delta_r \setminus \Delta_{r_A}) \cup \Delta_{e_B}$ by combining the perturbation values of pixels from the set $(M_r \setminus A)$ with the values of pixels B from the opposing solution.

The crossover operator is implemented as another sampling method, where the distribution is constructed from a solution within the population. We provide the pseudo-code in Algorithm 5 within the appendix.

Mutation. The purpose of the mutation operator is to explore promising areas by introducing variation to solutions generated by the crossover operator. In the Sparse-RS attack algorithm developed by Croce et al. [10], the variation operator was shown to be a powerful method for conducting local search. We have adapted a similar operator for our scenario.

To perform variation on a solution P'_a, we randomly modify both its pixel locations M'_a and perturbation values Δ'_a. We begin by defining a set of all pixel locations $U = \{1, \cdots, h \cdot w\}$ and remove any overlapping locations with M'_a to obtain $T = U \setminus M'_a$. We then randomly select two sets $A \subset M'_a$ and $B \subset T$ with $|A| = |B| = p_m \cdot k$ where p_m is a user-defined parameter that determines the percentage of pixels to change. We generate the output $M''_a = (M'_a \setminus A) \cup B$ by replacing the pixels in A with those in B. The perturbation values Δ_B for the pixel locations in B are generated by sampling from the set $\{-1, 1, 0\}$ for each color channel, with the probability of choosing 0 being the same as the initialization probability pr_0. Finally, we combine Δ''_a with the values of the set of unchanged pixel locations to obtain the mutated solution. Algorithm 6 within the appendix describes the mutation operator.

Evaluation. To evaluate a solution with pixel locations M_s and perturbation values Δ_s, the perturbation $\vec{\delta_s}$ is first initialized as a zero-matrix with the same shape as the attacked image $\mathbf{x}$. Then, for each pixel location in M_s, the corresponding value in $\vec{\delta_s}$ is set to the value in Δ_s. The adversarial image is then constructed by adding $\vec{\delta_s}$ to the original image $\mathbf{x}$. Finally, the objective vector defined in equation (5) is used to evaluate the solution.

Selection. The selection operator determines which individuals in the sets P and O of evaluated parent and offspring solutions, respectively, are better and should be placed within the population of the next iteration.

In our multi-objective context, we use non-dominated sorting [13] on the combined population $P \cup O$ to generate a series of subsets called $fronts$. Solutions within each $front$ cannot be determined to be better or worse than other solutions within the same $front$. Solutions within the first $front$ are dominated by the smallest number of solutions from P and are considered the better solutions in the combined population. Solutions in the last $front$ are dominated by the largest number of solutions and are considered the worst performing solutions. The $front$ to which each solution belongs is referred to as its rank. To construct the population of the next generation, we select the s solutions with the lowest rank. We present the pseudo-code for the non-dominated sorting method in Algorithm 1 within the appendix.

To achieve our aim of constructing an adversarial perturbation while also minimizing its l_2 norm, we propose the following domination to be used within the non-dominated sorting method:

Definition 3.1 (Domination)**.** *Given two solutions P_i and P_j from the set P that produce perturbations $\vec{\delta}_i$ and $\vec{\delta}_j$ which are evaluated by equation (5) to produce objective vectors $\mathbf{F}_i$ and $\mathbf{F}_j$, P_i is said to dominate P_j if one of the following conditions are satisfied:*

- *$\vec{\delta}_i$ is adversarial whereas $\vec{\delta}_j$ is not.*
- *Both $\vec{\delta}_i$ and $\vec{\delta}_j$ are adversarial and $\|\vec{\delta}_i\|_2 < \|\vec{\delta}_j\|_2$.*
- *Both $\vec{\delta}_i$ and $\vec{\delta}_j$ are **not** adversarial and $\mathcal{L}(\vec{\delta}_i) < \mathcal{L}(\vec{\delta}_j)$.*

Note that a perturbation $\vec{\delta}$ is said to be *adversarial* in the case either equation (1) or equation (3) is met, the choice of loss function $\mathcal{L}(\cdot)$ is dependent on the type of attack.

This domination relation encourages our attack method to generate perturbation values with smaller l_2 norms, but not at the expense of being *adversarial*.

4. Experiments

In this section, we present our empirical evaluation of the proposed method's effectiveness by attacking models trained on the CIFAR-10 [25] and ImageNet [14] datasets. Prior to describing the experiments in detail, we outline the experimental setup in Section 4.1. To improve the design of our method, we conduct an ablation study in Section 4.2 to determine the relative importance of its various components. Finally, we compare the proposed method with state-of-the-art sparse adversarial attacks of GreedyFool [15], SAPF [16], Homotopy Attack [47], and a Sparse-RS [10] method that we adapt to our scenario, in Section 4.3. Additionally, we compare the performance of the proposed method with conventional domination relation [13].

4.1. Experiment Setup

Dataset and Model Settings: In this work, we evaluate our method on two datasets: CIFAR-10 and ImageNet. For CIFAR-10, we use 1000 correctly classified images from the test set to conduct non-targeted and targeted attacks on two adversarial trained classifiers, AT$_1$ [19] and AT$_2$ [20], as well as a conventionally trained classifier, provided within the RobustBench library [9], with and without the black-box RND defense mechanism [33]. We conduct targeted attacks with five different target classes per image. Despite targeted attacks being more difficult than non-targeted attacks, we set the maximum number of model queries to 1000 for all CIFAR-10 attacks.

For ImageNet, we attack four classifiers: Efficient Convolutional Neural Network (MobileNet) [22], Deep Residual Network (ResNet50) [21], and two adversarial trained classifiers, TL$_1$ and TL$_2$ [35]. We also attack the MobileNet classifier with RND defense [33]. For each model, we randomly select 1000 correctly classified images from the ImageNet validation set and conduct non-targeted attacks with a budget of 5000 queries due to the larger size of the images.

All adversarial trained classifiers were implemented using the RobustBench [9] library.

Parameter Setting: For targeted attacks we make use of the cross-entropy loss of the target class y_t,

$$\mathcal{L}(f;,\mathbf{x}+\vec{\delta},y_t) = -f_{y_t} + \log(\sum_{i=1}^{K} e^{f_i}) \tag{6}$$

whereas for non-targeted attacks we employ the margin loss

$$\mathcal{L}(f;,\mathbf{x}+\vec{\delta},y_q) = f_y - f_{y_q} \tag{7}$$

where y and y_q are defined in equation (2).

As described in Section 3, our method constrains the perturbation values to the set $\{-1,1,0\}$, in order to provide a fair comparison with other algorithms, we allow the l_∞ constraint of GreedyFool [15] and Homotopy Attack [47] to be unbounded also. The compared SAPF algorithm [16] does not constrain the l_∞ norm of generated perturbations, therefore, we make no changes to its original implementation.

We implement each compared algorithm using the authors' original implementation. To set the value of the initial sparsity k, we follow the work of Croce et al. [10] and set $k = 24$ for the CIFAR-10 dataset and $k = 150$ for the ImageNet dataset, which correspond to 2.34% and 0.30% of the total number of pixels, respectively. We keep the hyper-parameters of the adapted Sparse-RS algorithm constant with those used in its original implementation for targeted and non-targeted attacks. We set the population size $s = 2$ and the zero-sample probability $pr_0 = 0.3$ of the proposed method for all conducted experiments.

Evaluation Metrics: We evaluate the performance of all algorithms by allowing them to exhaust the query budget when attacking each classifier. We report the average l_p-norm of generated adversarial perturbations, where $p = 0$ and $p = 2$. Additionally, we report the attack success rates (ASR) and the average structural similarity index measure (SSIM) for each method when attacking all CIFAR-10 and ImageNet models. The SSIM is used to measure the similarity between the original image and the corresponding adversarial example, with values closer to 1 indicating higher similarity.

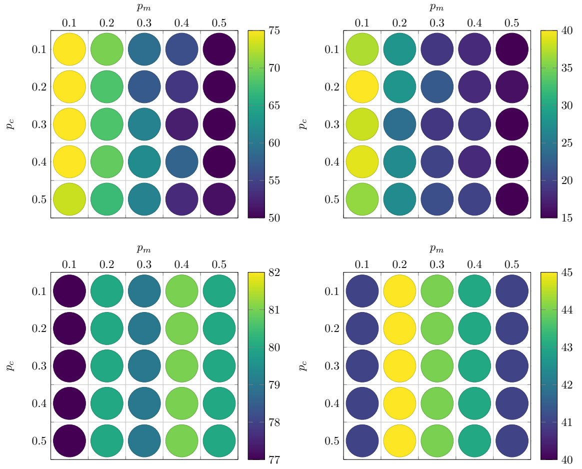

Figure 2. Correlation plots showing the average success rate of each p_m, p_c configuration for targeted (right) and non-targeted (left) attacks on CIFAR-10 images, the top row corresponds to the static approach where the bottom row corresponds to the dynamic approach of Croce et al. [10].

4.2. Ablation Study

To evaluate the contribution of different mechanisms in our proposed method, we conduct an ablation study by attacking models trained on the CIFAR-10 dataset. Specifically, we vary the values of two key parameters: the probability of crossover (p_c) and the probability of mutation (p_m). We conduct a grid search over the values of p_c and p_m, with each value ranging from 0.1 to 0.5 in increments of 0.1.

For each configuration of p_c and p_m, we randomly select 100 images from the CIFAR-10 test set and attack the AT$_1$ model under both the targeted and non-targeted attack scenarios. We report the results in terms of attack success rate (ASR), l_p-norm of the generated perturbations (where $p = 0$ and $p = 2$), and average structural similarity index measure (ssim).

Note that for each configuration, we keep the remaining hyperparameters of our method fixed with those used in the original implementation, including the initial sparsity (k), population size (s), and zero-sample probability (pr_0).

Mutation Operator: The effectiveness of the mutation operator depends on the value of the parameter p_m, which determines the magnitude of the mutation. Croce et al. [10] proposed a dynamic method that decreases p_m as the attack process progresses, as shown in Algorithm 2 in the appendix. In contrast, many evolutionary approaches keep the hyper-parameters, including p_m, constant during optimization [12, 13, 39]. To determine the optimal method for selecting p_m, we compared the performance of the proposed attack using both approaches. We see the top-performing p_m, p_c configurations in both approaches achieve similar success rates in Fig. 2. However, the dynamic method proposed by Croce et al. [10] is more reliable in achieving higher success rates across different values of p_m. Additionally, Fig. 2 shows that changes to p_m significantly impact the performance of both approaches, while changes to p_c have minimal effect. This is expected since smaller population sizes, such as $s = 2$, have less variation, resulting in faster convergence. Once converged, the mutation operator alone navigates the search space, highlighting the importance of the p_m parameter.

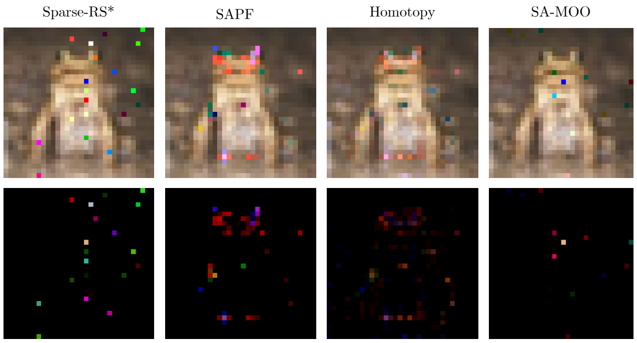

Figure 3. Adversarial images and corresponding perturbations generated by sparse attack methods conducting targeted attacks on the AT$_1$ model. The ground truth is Frog and the target label is Dog.

Crossover Operator: We evaluated the importance of the crossover operator with $s = 2$ by comparing the performance of the proposed method with and without it when attacking both the AT$_1$ and AT$_2$ models. The results presented in Table 4 in the appendix show that the performance of the proposed method is unaffected by the exclusion of the crossover operator, therefore we keep the crossover operator in our attack method for all subsequent experiments.

Due to the superior performance shown in Fig. 2, for targeted attacks, we set $p_m = 0.2$ and $p_c = 0.1$, and for non-targeted attacks, we set $p_m = 0.4$ and $p_c = 0.1$ for the rest of this paper.

4.3. Comparison

To evaluate the performance of the proposed method we compare with the state-of-the-art (SOTA) white-box sparse adversarial attacks GreedyFool [15], SAPF [16] and Homotopy Attack [47]. Due to the lack of black-box sparse attack methods that minimize the l_0 and l_2 norms of the perturbation, we adapt the state-of-the-art Sparse-RS [10] algorithm by replacing its solution loss comparison with our domination mechanism in Definition 3.1 and allowing a 0 perturbation value to be sampled at a probability of pr_0. We give full detail of the adapted Sparse-RS method in Algorithm 3 within the appendix.

	AT$_1$				AT$_2$			
Method	ASR	l_0	l_2	SSIM	ASR	l_0	l_2	SSIM
SA-MOO	**84.40%**	**15.02**	8.00	0.95	**76.90%**	**15.28**	8.54	0.95
SA-MOO*	82.50%	19.64	8.07	0.95	74.00%	19.60	9.05	0.95
Sparse-RS*	83.50%	21.10	14.33	0.92	76.10%	20.94	14.60	0.91
SAPF	49.20%	57.83	9.94	0.93	52.40%	61.23	11.20	0.93
Homotopy	23.20%	73.30	4.16	0.95	34.90%	118.30	7.15	0.95
GreedyFool	16.50%	999.96	**0.10**	**0.99**	13.40%	1000.53	**0.11**	**0.99**

	AT$_1$				AT$_2$			
Method	ASR	l_0	l_2	SSIM	ASR	l_0	l_2	SSIM
SA-MOO	**44.20%**	**18.37**	11.92	0.93	**39.10%**	**18.81**	12.90	0.93
SA-MOO*	8.60%	23.5	21.06	0.89	8.8%	23.268	19.32	0.89
Sparse-RS*	**44.20%**	21.17	14.37	0.91	**39.10%**	20.96	15.88	0.90
SAPF	7.40%	69.54	13.02	0.90	6.80%	65.21	11.64	0.91
Homotopy	13.80%	140.66	11.81	0.92	7.50%	164.25	10.90	0.93
GreedyFool	2.40%	1000.54	**0.10**	**0.99**	1.50%	1001.20	**0.11**	**0.98**

Table 1. Statistics of attack success rate, average SSIM and average l_p norms ($p = 0, 2$) of non-targeted (top) and targeted (bottom) attacks on adversarial trained CIFAR-10 classifiers.

	Non-Targeted				Targeted			
Method	ASR	l_0	l_2	SSIM	ASR	l_0	l_2	SSIM
SA-MOO	**91.5%**	**18.93**	7.81	0.95	**86.5%**	**18.99**	10.83	0.93
SA-MOO*	87.3%	20.196	15.36	0.96	9.00%	19.43	6.11	0.96
Sparse-RS*	90.1%	20.340	13.65	0.91	84.4%	21.877	15.02	0.90
SAPF	65.20%	104.54	28.31	0.85	27.60%	114.99	28.99	0.84
Homotopy	35.02%	52.55	3.89	0.94	27.45	130.26	10.91	0.94
GreedyFool	56.50%	1016.05	**0.16**	**0.99**	36.89	1016.05	**0.16**	**0.99**

	Non-Targeted				Targeted			
Method	ASR	l_0	l_2	SSIM	ASR	l_0	l_2	SSIM
SA-MOO	**85.4%**	**17.60**	**7.16**	**0.95**	**83.3%**	**18.28**	**10.55**	**0.92**
SA-MOO*	84.3%	23.04	17.29	0.89	8.40%	23.02	17.19	0.89
Sparse-RS*	84.2%	22.460	16.78	0.93	83.1%	23.67	16.23	0.93

Table 2. Statistics of attack success rate, average SSIM and average l_p norms ($p = 0, 2$) of attacks on conventionally trained CIFAR-10 classifier with (bottom) and without (top) the RND black-box defense mechanism.

CIFAR-10: Table 1 presents the results of targeted and non-targeted attacks on adversarial trained classifiers. The proposed SA-MOO method and the adapted Sparse-RS (Sparse-RS*) method achieve similar attack success rates, but the SA-MOO method is better at reducing the l_0 and l_2 norms of the perturbation, as reflected by the average structural similarity (SSIM) achieved. The proposed method also generates adversarial perturbations with similar l_2 norms as white-box attack methods, but modifies a much smaller number of pixels. The GreedyFool algorithm generates adversarial perturbations with a smaller l_2 norm, but its low success rates demonstrate its limited capability under highly constrained conditions.

Comparing the performance of the proposed SA-MOO method with the conventional domination relation (SA-MOO*) shows similar performance for non-targeted attacks, but the use of the conventional relation causes the performance of the SA-MOO method to deteriorate for targeted attacks. This demonstrates the importance of appropriate handling of the multi-objective attack scenario.

Table 2 presents the attack statistics for a CIFAR-10 classifier trained with conventional methods. We observe that all attacks achieve higher success rates compared to adversarial trained classifiers, especially in the case of targeted attacks. This suggests that adversarial training is a promising defense mechanism against targeted attacks, but still leaves the classifier vulnerable to non-targeted attacks. Despite the RND method being developed to defense against black-box attacks, 2 indicates only a slight decrease in performance compared to the classifier without the defense mechanism.

ImageNet: Table 3 displays the results of non-targeted attacks on classifiers trained on the ImageNet dataset. Similar to the results for CIFAR-10 classifiers, the proposed method achieves a higher success rate compared to its competitors. Furthermore, the proposed method is able to generate adversarial perturbations with similar l_2 norms while

	TL$_1$				TL$_2$			
Method	ASR	l_0	l_2	SSIM	ASR	l_0	l_2	SSIM
SA-MOO	**99.20%**	**134.00**	45.25	**0.99**	**99.20%**	**133.83**	45.47	**0.99**
SA-MOO*	99.13	135.73	67.61	0.98	99.08	135.46	67.42	0.98
Sparse-RS*	97.20%	146.72	65.89	0.98	97.60%	142.28	66.82	0.98
SAPF	93.10%	3370.83	52.83	0.98	94.20%	3220.62	51.07	0.98
Homotopy	89.72%	769.02	45.87	0.98	91.28%	749.01	68.92	0.98
GreedyFool	45.25%	1002.53	**5.22**	**0.99**	47.37%	993.14	**5.02**	**0.99**

	MobileNet				ResNet			
Method	ASR	l_0	l_2	SSIM	ASR	l_0	l_2	SSIM
SA-MOO	**98.24%**	**113.44**	40.51	**0.99**	**97.58%**	**130.61**	38.93	**0.99**
SA-MOO*	98.20%	133.86	68.48	0.98	86.27%	133.84	68.39	0.98
Sparse-RS*	97.69%	135.10	60.10	0.98	96.88%	134.39	59.12	0.98
SAPF	91.04%	3840.39	55.26	0.93	92.72%	4733.75	47.72	0.86
Homotopy	87.48%	802.42	42.03	0.95	86.42%	905.81	40.62	0.94
GreedyFool	38.82%	1127.0	**5.92**	**0.99**	41.18%	1721.42	**5.95**	**0.99**

	MobileNet				ResNet			
Method	ASR	l_0	l_2	SSIM	ASR	l_0	l_2	SSIM
SA-MOO	**98.33%**	**139.36**	**91.26**	**0.98**	**97.22%**	**142.82**	**95.72**	**0.98**
SA-MOO*	98.25%	139.79	91.59	0.98	96.92%	143.87	97.32	0.98
Sparse-RS*	96.28%	145.12	100.42	0.97	95.55%	148.29	102.45	0.97

Table 3. Statistics of attack success rate, average SSIM and average l_p norms ($p = 0, 2$) of non-targeted attacks on adversarial trained ImageNet classifiers (top) and conventionally trained ImageNet classifiers with (bottom) and without (middle) the RND black-box defense mechanism.

perturbing fewer pixels.

The success rate and l_0 norm of the proposed method with the conventional domination relation are highly similar to those achieved using the proposed domination relation in Definition 3.1. However, the average l_2 norm of the perturbation for adversarial and conventionally trained classifiers is much lower for the proposed method with domination relation (3.1). This once again demonstrates the impact that a solution comparison mechanism such as the domination relation can have when addressing a multi-objective task. We provide a comparison of adversarial images generated by the proposed method using both domination relations in Figure 4.

5. Conclusion and Future Directions

The crux of SA-MOO is a multi-objectivized way (a bi-objective optimization of the loss and the l_2 norm) to generate adversarial examples. Instead of simply weighted-aggregating different objectives, SA-MOO uses a population-based meta-heuristic to search for a set of trade-off alternatives. A novel dominance relation is proposed to help prioritize adversarial example generation over reducing the corresponding modifications. It's usefulness is validated by comparing with a conventional domination relation and the Sparse-RS algorithm. Though we focus on black-box adversarial attacks, the framework is adaptable to white-box attacks (e.g., by using multiple gradient descent) in case the gradient information is accessible. Last but not least, the proposed dominance relation empowers the decision-maker to tweak the importance of different objectives, offering a pathway towards explainable and controllable decision-making.

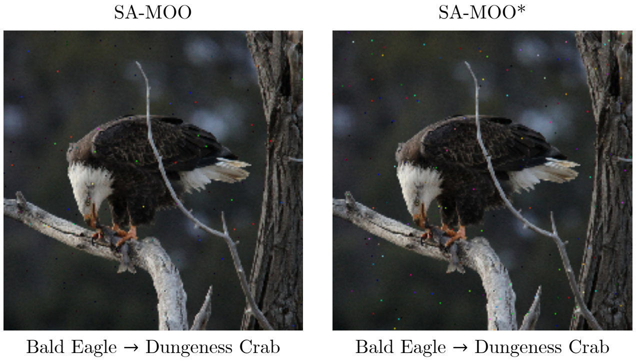

Figure 4. Adversarial Images generated by the proposed method with conventional domination relation (SA-MOO*) and domination relation defined in Definition 3.1 (SA-MOO) when attacking MobileNet ImageNet classifier.

6. Acknowledgement

This work was supported by UKRI Future Leaders Fellowship (MR/S017062/1 and MR/X011135/1), EPSRC (2404317), NSFC (62076056), Royal Society (IES/R2/212077) and Amazon Research Award.

References

[1] Moustafa Alzantot, Yash Sharma, Supriyo Chakraborty, and Mani B. Srivastava. Genattack: Practical black-box attacks with gradient-free optimization. *CoRR*, 2018.

[2] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: A query-efficient black-box adversarial attack via random search. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXIII*. Springer, 2020.

[3] Tao Bai, Jinqi Luo, Jun Zhao, Bihan Wen, and Qian Wang. Recent advances in adversarial training for adversarial robustness. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, 2021.

[4] Arjun Nitin Bhagoji, Warren He, Bo Li, and Dawn Song. Practical black-box attacks on deep neural networks using efficient query mechanisms. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XII*, Lecture Notes in Computer Science, 2018.

[5] Stephen P. Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 2011.

[6] Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*. IEEE Computer Society, 2017.

[7] Pin-Yu Chen, Yash Sharma, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. EAD: elastic-net attacks to deep neural networks via adversarial examples. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, 2018.

[8] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. ZOO: zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, AISec@CCS 2017, Dallas, TX, USA, November 3, 2017*, 2017.

[9] Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021.

[10] Francesco Croce, Maksym Andriushchenko, Naman D. Singh, Nicolas Flammarion, and Matthias Hein. Sparse-rs: A versatile framework for query-efficient sparse black-box adversarial attacks. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022*. AAAI Press, 2022.

[11] Francesco Croce and Matthias Hein. Sparse and imperceivable adversarial attacks. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, 2019.

[12] Kalyanmoy Deb. *Multi-objective optimization using evolutionary algorithms*. Wiley-Interscience series in systems and optimization. Wiley, 2001.

[13] Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.*, 2002.

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

[15] Xiaoyi Dong, Dongdong Chen, Jianmin Bao, Chuan Qin, Lu Yuan, Weiming Zhang, Nenghai Yu, and Dong Chen. Greedyfool: Distortion-aware sparse adversarial attack. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[16] Yanbo Fan, Baoyuan Wu, Tuanhui Li, Yong Zhang, Mingyang Li, Zhifeng Li, and Yujiu Yang. Sparse adversarial attack via perturbation factorization. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXII*, 2020.

[17] Xianghong Fang, Haoli Bai, Ziyi Guo, Bin Shen, Steven C. H. Hoi, and Zenglin Xu. DART: domain-adversarial residual-transfer networks for unsupervised cross-domain image classification. *Neural Networks*, 2020.

[18] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[19] Sven Gowal, Chongli Qin, Jonathan Uesato, Timothy A. Mann, and Pushmeet Kohli. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *CoRR*, 2020.

[20] Sven Gowal, Sylvestre-Alvise Rebuffi, Olivia Wiles, Florian Stimberg, Dan Andrei Calian, and Timothy A. Mann. Improving robustness using generated data. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, 2021.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV*, Lecture Notes in Computer Science. Springer, 2016.

[22] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, 2017.

[23] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In *Proceedings of the 35th Interna-*

tional Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018, Proceedings of Machine Learning Research. PMLR, 2018.

[24] Francisco Erivaldo Fernandes Junior and Gary G. Yen. Particle swarm optimization of deep neural networks architectures for image classification. *Swarm Evol. Comput.*, 2019.

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

[26] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *CoRR*, 2016.

[27] Mark Lee and J. Zico Kolter. On physical adversarial patches for object detection. *CoRR*, 2019.

[28] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[29] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *CoRR*, 2017.

[30] Nina Narodytska and Shiva Prasad Kasiviswanathan. Simple black-box adversarial attacks on deep neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society, 2017.

[31] Nicolas Papernot, Patrick D. McDaniel, and Ian J. Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *CoRR*, 2016.

[32] Nicolas Papernot, Patrick D. McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *IEEE European Symposium on Security and Privacy, EuroS&P 2016, Saarbrücken, Germany, March 21-24, 2016*, 2016.

[33] Zeyu Qin, Yanbo Fan, Hongyuan Zha, and Baoyuan Wu. Random noise defense against query-based black-box attacks. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, 2021.

[34] Binxin Ru, Adam D. Cobb, Arno Blaas, and Yarin Gal. Bayesopt adversarial attack. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020.

[35] Hadi Salman, Andrew Ilyas, Logan Engstrom, Ashish Kapoor, and Aleksander Madry. Do adversarially robust imagenet models transfer better? In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[36] Yash Sharma and Pin-Yu Chen. Attacking the madry defense model with l_1-based adversarial examples. *CoRR*, abs/1710.10733, 2017.

[37] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[38] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin A. Riedmiller. Striving for simplicity: The all convolutional net. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*, 2015.

[39] Rainer Storn. Differential evolution design of an iir-filter. In *Proceedings of 1996 IEEE International Conference on Evolutionary Computation, Nayoya University, Japan, May 20-22, 1996*. IEEE, 1996.

[40] Rainer Storn and Kenneth V. Price. Differential evolution - A simple and efficient heuristic for global optimization over continuous spaces. 1997.

[41] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Trans. Evol. Comput.*, 2019.

[42] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[43] Simen Thys, Wiebe Van Ranst, and Toon Goedemé. Fooling automated surveillance cameras: Adversarial patches to attack person detection. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2019*. Computer Vision Foundation / IEEE, 2019.

[44] Ye Tian, Jingwen Pan, Shangshang Yang, Xingyi Zhang, Shuping He, and Yaochu Jin. Imperceptible and sparse adversarial attacks via a dual-population based constrained evolutionary algorithm. *IEEE Transactions on Artificial Intelligence*, 2022.

[45] Chun-Chen Tu, Pai-Shun Ting, Pin-Yu Chen, Sijia Liu, Huan Zhang, Jinfeng Yi, Cho-Jui Hsieh, and Shin-Ming Cheng. Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, 2019.

[46] Jonathan Uesato, Brendan O'Donoghue, Pushmeet Kohli, and Aäron van den Oord. Adversarial risk and the dangers of evaluating against weak attacks. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, Proceedings of Machine Learning Research, 2018.

[47] Mingkang Zhu, Tianlong Chen, and Zhangyang Wang. Sparse and imperceptible adversarial attack via a homotopy algorithm. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, Proceedings of Machine Learning Research. PMLR, 2021.

[48] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. Learning transferable architectures for scalable image recognition. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT,*

USA, June 18-22, 2018. Computer Vision Foundation / IEEE Computer Society, 2018.

7. Appendix

Algorithm 1: Non-Dominating Sorting

```
Input: combined population $P$, objective vectors $L$
$\mathcal{F} \leftarrow \{\}$ // set of fronts
for $p \in P$ do
    $S_p \leftarrow \{\}$ //set of $p$ dominated solutions
    $n_p \leftarrow 0$ // domination counter of $p$
    for $q \in P$ do
        if $p$ dominates $q$ then
            $S_p \leftarrow S_p \cup \{q\}$
        else if $q$ dominates $p$ then
            $n_p \leftarrow n_p + 1$
    if $n_p == 0$ then
        $p_{rank}$ = 1 // $p$ belongs to the first front
        $\mathcal{F}_1 \leftarrow \mathcal{F}_1 \cup \{p\}$
    $i \leftarrow 1$ // initialize front counter
    while $\mathcal{F}_i \neq \emptyset$ do
        $Q \leftarrow \emptyset$ // store solutions of the next front
        for $p \in \mathcal{F}_i$ do
            for $q \in S_p$ do
                $n_q \leftarrow n_q - 1$
                if $n_q == 0$ then
                    // $q$ belongs to the next front
                    $q_{rank} \leftarrow i + 1$
                    $Q \leftarrow Q \cup \{q\}$
        $i \leftarrow i + 1$
        $\mathcal{F}_i \leftarrow Q$
    return $\mathcal{F}$
```

Algorithm 1 described the non-dominated sorting method proposed by Deb et al. [13] for sorting solutions with multiple objectives. The method first determines solutions in the first front (not dominated by any solutions) then iteratively constructs the remaining fronts.

Algorithm 2: Sparse-RS p_m Selection Method

```
Input: conducted function evaluations $i$, budget $N$,
       initial mutation size $\alpha_{init}$
$t \leftarrow$ int($\frac{i}{N} \cdot 10000$)
$c \leftarrow \{0, 50, 200, 500, 1000, 2000, 4000, 6000, 8000\}$
$j \leftarrow$ index of $c$ that is closest to $t$
$\beta \leftarrow \{2, 4, 5, 6, 8, 10, 12, 15, 20\}$
return $\alpha_{init}/\beta_j$
```

Algorithm 2 reduces the mutation size as the number of classifiers queries increases. The method linearly re-scales the current number of models i with the assumption of $N = 10000$.

Algorithm 3: Adapted Sparse-RS Attack

```
Input: objective vector $F$, input $\mathbf{x} \in \mathcal{X}$, sparsity $k$,
       zero-sampling $pr_0$, initial mutation size
       $\alpha_{init}$, budget $N$
$M \leftarrow k$ random pixel indices to be perturbed
$\Delta \leftarrow$ values of the perturbation to be applied
$L \leftarrow F(\mathbf{x}; \{M, \Delta\})$
for $i \leftarrow 0; i < N; i + 1$ do
    $p_m \leftarrow selection(\alpha_{init})$ // refer to Algorithm 2
    $M', \Delta' \leftarrow mutation(\{M, \Delta\}, p_m)$
    $L' \leftarrow F(\mathbf{x}; \{M', \Delta'\})$
    if $\{M', \Delta'\}$ dominates $\{M, \Delta\}$ then
        $M \leftarrow M', \Delta \leftarrow \Delta', L \leftarrow L'$
return $\{M, \Delta\}$
```

Algorithm 3 outlines the Sparse-RS algorithm proposed by Croce at al. [10] adapted to the multi-objective scenario. *dominates* is corresponds to Definition 3.1.

Algorithm 4: SA-MOO Method

```
Input: objective vector $F$, input $\mathbf{x} \in \mathcal{X}$, query
       budget $N$, sparsity $k$, population size $s$,
       zero-sampling probability $pr_0$
// Initial Population
$P \leftarrow \{\{M_1, \Delta_1\}, \cdots, \{M_s, \Delta_s\}\}$
// Objective Evaluation
$L \leftarrow \{F(\mathbf{x}; \{M_1, \Delta_1\}), \cdots, F(\mathbf{x}; \{M_s, \Delta_s\})\}$
for $i \leftarrow 0;\ i < N;\ i \leftarrow i + s$ do
    // Uniformly Sample $s/2$ pairs of $P$ indices
    $J \leftarrow \mathcal{U}(\{1, \cdots, s\})^{\frac{s}{2} \times 2}$
    $P_O \leftarrow \{\}$
    $L_O \leftarrow \{\}$
    for $j \in J$ do
        $O \leftarrow crossover(P_{j_0}, P_{j_1})$
        $M_1'', \Delta_1'' \leftarrow mutation(O_1)$
        $M_2'', \Delta_2'' \leftarrow mutation(O_2)$
        $P_O \leftarrow P_O \cup \{\{M_1'', \Delta_1\}, \{M_2'', \Delta_2\}\}$
        $L_O \leftarrow L_O \cup \{F(\mathbf{x}; \{M_1'', \Delta_1''\})\}$
        $L_O \leftarrow L_O \cup \{F(\mathbf{x}; \{M_2'', \Delta_2''\})\}$
    $P \leftarrow P \cup P_O$
    $L \leftarrow L \cup L_O$
    $P \leftarrow$ non-dominated sorting$(P)$
    $P \leftarrow P_{1:s}$ // Select lowest ranked solutions
    $L \leftarrow L_P$
return $P$ // return population of solutions
```

	AT$_1$				AT$_2$			
Method	ASR	l_0	l_2	SSIM	ASR	l_0	l_2	SSIM
SA-MOO*	**84.40%**	**15.02**	**8.00**	**0.95**	**76.90%**	**15.28**	**8.54**	**0.95**
SA-MOO**	**84.40%**	**15.02**	**8.00**	**0.95**	**76.90%**	**15.28**	**8.54**	**0.95**
SA-MOO*	**44.20%**	18.39	11.93	**0.93**	**39.10%**	**18.81**	**12.90**	**0.93**
SA-MOO**	**44.20%**	**18.37**	**11.92**	**0.93**	**39.10%**	18.82	**12.90**	**0.93**

Table 4. Statistics of attack success rate, average ssim, and average l_0, l_2 distances of non-targeted (top) and targeted (bottom) attacks on the CIFAR-10 trained models AT$_1$ and AT$_2$. Where "SA-MOO" is the proposed method, ** refers to both *crossover* and *mutation* operators, * refers to only the *mutation operator.*

Algorithm 5: Crossover Operator

```
Input: pixel locations $M_a$, $M_b$, perturbation values
       $\Delta_a, \Delta_b$, crossover size $p_c$, sparsity $k$
$O \leftarrow \{\}$
for $r \in \{a, b\}$ and $e \in \{b, a\}$ do
    $U \leftarrow M_e \setminus (M_r \cap M_e)$
    $b \leftarrow \min\{p_c \cdot k, |U|\}$
    $A \leftarrow \mathcal{U}(M_r)^b$
    $B \leftarrow \mathcal{U}(U)^b$
    $M'_r \leftarrow (M_r \setminus A) \cup B$
    $\Delta'_r \leftarrow (\Delta_r \setminus \Delta_{r_A}) \cup \Delta_{e_B}$
    $O \leftarrow O \cup \{\{M'_r, \Delta'_r\}\}$
return $O$
```

Algorithm 6: Mutation Operator

```
Input: pixel locations $M'_a$, perturbation values $\Delta'_a$,
       mutation size $p_m$, sparsity $k$, zero-sample
       probability $pr_0$
$U \leftarrow \{1, \cdots, h \cdot w\}$ // image height $h$ and width $w$
$T \leftarrow U \setminus M'_a$
$A \leftarrow \mathcal{U}(M'_a)^{p_m \cdot k}$
$B \leftarrow \mathcal{U}(T)^{p_m \cdot k}$
$M''_a \leftarrow (M'_a \setminus A) \cup B$
// Sample with a zero-probability $pr_0$
$\Delta''_a \leftarrow (\Delta'_a \setminus \Delta'_{a_A}) \cup \mathcal{U}(\{-1, 0, 1\})^{(p_m \cdot k) \times 3}$
return $\{M''_a, \Delta''_a\}$
```