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.

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 $l_\infty$ 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 $l_\infty$-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 attack 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 $\infty$. 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 $l_\infty$ 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 $l_\infty$ 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 Madry et al. [29] to the $l_0$ ball. They also utilize a heuristic search method called CornerSearch that searched over all pixels and selects subset to perturb. The SAPF algorithm of Fan et al. [16], formulates the sparse adversarial attack as a mixed integer programming task and applied 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 methods. The method first selects perturbations with large gradients, then removes those that do not impact the desired misclassification. The Homotopy attack algorithms 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 $l_{\infty}$ 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 $f : X \subseteq [0, 1]^{h \times w \times 3} \rightarrow \mathbb{R}^K$ be a trained DNN image classifier that takes a benign RGB image $x \in X$ of height $h$ and width $w$ as its input and outputs a label $y = \arg \max_{i \in \{1, \cdots, K\}} f_i(x + \delta)$, where $K$ is the number of class labels.

A non-targeted attack aims to search for a perturbation $\delta \in \mathbb{R}^{h \times w \times 3}$ for $x$ such that:

$$\arg \max_{i \in \{1, \cdots, K\}} f_i(x + \delta) \neq y, \quad (1)$$

where $y$ is the ground truth class label for $x$. 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:

$$\min_{\delta} L(f; x + \delta, y_q) \quad \text{s.t. } ||\delta||_0 \leq \epsilon, \ 0 \leq x + \delta \leq 1 \quad (2)$$

where $y_q = \arg \max_{q \neq y} f_q(x)$ and the minimization of the loss function $L$, 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

$$\arg \max_{i \in \{1, \cdots, K\}} f_i(x + \delta) = y_t, \quad (3)$$

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,

$$\min_{\delta} L(f; x + \delta, y_t) \quad \text{s.t. } ||\delta||_0 \leq \epsilon, \ 0 \leq x + \delta \leq 1. \quad (4)$$

Existing algorithms in the literature typically focus on solving equation (2) and equation (4) equation (2) and equation (4) by either fixing the number of perturbed pixel to a specific value $\epsilon$ and allowing unbounded modifications or constraining its $l_2$ or $l_{\infty}$ 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 difficult of minimizing the $l_0$ norm of an adversarial perturbation by greedily removing perturbed pixel. Therefore, we argue than an improved methods should jointly search for perturbations with small $l_0$ and $l_2$ norms while minimizing the loss function $L(x + \delta)$. Our aim is to generate a $\delta$ that solves the following optimization problem:

$$\min_{\delta} F(x + \delta) \quad \text{s.t. } ||\delta||_0 \leq \epsilon, \ 0 \leq x + \delta \leq 1, \quad (5)$$

where $F(x + \delta) = (L(x + \delta), ||\delta||_2, ||\delta||_0)^T$ is the objective vector. Here, $L(\cdot)$ 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 $l_0$ norm of the perturbation approached 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 $F(x + δ) = (L(x + δ), ||δ||_2)^T$, where $δ \in \{-1, 0, 1\}$.

### 3.2. Multi-Objective Optimization

Evolutionary algorithms are optimization 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 $\Delta_a, \Delta_b$, the crossover operator produces two new solutions that combine the traits 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 $|A| = |B| = \min p_e \cdot k, |U|$, where $k$ is the maximum number of modified pixels and $p_e$ 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 $\Delta'r$ 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 $P'a$, which is generated by the crossover operator and has pixel locations $M'a$ and perturbation values $\Delta'a$, we utilize the mutation operator shown in Algorithm 6. This operator produces a single solution by randomly modifying both $M'a$ and $\Delta'a$. 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 $\Delta''_a$ 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 $\Delta''_a$ with the values of the set of unchanged pixel locations to obtain the mutated solution.

**Evaluation.** To evaluate a solution with pixel locations $M_s$ and perturbation values $\Delta_s$, the perturbation $\delta$ is first initialized as a zero-matrix with the same shape as the attacked image $x$. Then, for each pixel location in $M_s$, the corresponding value in $\delta$ is set to the value in $\Delta_s$. The adversarial image is then constructed by adding $\delta$ to the original image $x$. 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 have a higher chance of passing on their genetic material to the next generation.

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 primary aim of constructing an adversarial perturbation while also minimizing its $L_2$ norm, we propose a modified domination relation for the non-dominated sorting method. To this end, we achieve this we propose the following domination relation:

**Definition 3.1 (Domination).** Given two solutions $P_i$ and $P_j$ from set $P$ that produce perturbations $\delta_i$ and $\delta_j$ which are evaluated by equation (5) to produce objective vectors $F_i$ and $F_j$, $P_i$ is said to dominate $P_j$ if one of the following conditions are satisfied:

- $\delta_i$ is adversarial whereas $\delta_j$ is not.
- Both $\delta_i$ and $\delta_j$ are adversarial and $\|\delta_i\|_2 < \|\delta_j\|_2$.
- Both $\delta_i$ and $\delta_j$ are not adversarial and $\mathcal{L}(\delta_i) < \mathcal{L}(\delta_j)$.

Note that a perturbation $\delta$ is said to be adversarial in the case either equation (1) or equation (3) is met, the choice of loss function $\mathcal{L}(\cdot)$ 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 make a compare the performance of the proposed method with conventional domination relation.

#### 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, AT1 [19] and AT2 [20], as well as a conventionally trained classifier 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, TL1 and TL2 [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 their larger size.

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 $y_t$.

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

(6)

whereas for non-targeted attacks we employ the margin loss

$$\mathcal{L}(f; \cdot, x + \delta, y_q) = f_y - f_{y_q}$$

(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 $L_\infty$ constraint of GreedyFool [15] and Homotopy Attack [47] to be unbounded as well. On the other hand, the compared SAPF algorithm [16] does not constrain the $L_\infty$ 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 2.4% and 0.23% 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 $pr_0 = 0.3$ for all conducted experiments.

We set the population size $s = 2$ and the zero-sample probability $pr_0 = 0.3$ 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 $l_p-$
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.

<table>
<thead>
<tr>
<th>Method</th>
<th>AT1 succeed</th>
<th>$l_0$</th>
<th>$l_2$</th>
<th>SSIM</th>
<th>AT2 succeed</th>
<th>$l_0$</th>
<th>$l_2$</th>
<th>SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>SA-MOO</td>
<td>84.40%</td>
<td>15.62</td>
<td>8.60</td>
<td>0.95</td>
<td>76.90%</td>
<td>15.28</td>
<td>8.54</td>
<td>0.95</td>
</tr>
<tr>
<td>Sparse-RS*</td>
<td>82.50%</td>
<td>19.64</td>
<td>8.07</td>
<td>0.95</td>
<td>74.00%</td>
<td>19.60</td>
<td>9.05</td>
<td>0.95</td>
</tr>
<tr>
<td>SAPF</td>
<td>85.50%</td>
<td>21.10</td>
<td>14.33</td>
<td>0.92</td>
<td>76.10%</td>
<td>20.94</td>
<td>14.60</td>
<td>0.91</td>
</tr>
<tr>
<td>Homotopy</td>
<td>49.20%</td>
<td>57.83</td>
<td>9.94</td>
<td>0.93</td>
<td>52.40%</td>
<td>61.23</td>
<td>11.20</td>
<td>0.93</td>
</tr>
<tr>
<td>GreedyFool</td>
<td>23.20%</td>
<td>73.30</td>
<td>4.16</td>
<td>0.95</td>
<td>34.90%</td>
<td>118.30</td>
<td>7.15</td>
<td>0.95</td>
</tr>
</tbody>
</table>

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 AT1 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 ($p_{r_0}$).

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 ($p_m$).

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 Figure 2. However, the dynamic method proposed by Croce et al. [10] is more reliable in achieving higher success rates. Additionally, Figure 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.
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.

<table>
<thead>
<tr>
<th>Method</th>
<th>Non-Targeted</th>
<th>Targeted</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ASR</td>
<td>$l_0$</td>
</tr>
<tr>
<td>SA-MOO</td>
<td>91.5%</td>
<td>18.93</td>
</tr>
<tr>
<td>SA-MOO*</td>
<td>87.5%</td>
<td>20.196</td>
</tr>
<tr>
<td>Sparse-RS*</td>
<td>90.1%</td>
<td>20.340</td>
</tr>
<tr>
<td>SAPF</td>
<td>65.20%</td>
<td>104.34</td>
</tr>
<tr>
<td>Homotopy</td>
<td>35.02%</td>
<td>52.55</td>
</tr>
<tr>
<td>GreedyFool</td>
<td>56.50%</td>
<td>1016.05</td>
</tr>
</tbody>
</table>

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 AT1, 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 Figure 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 $l_{0,2}$ 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.

CIFAR-10: Table 1 presents the results of targeted and non-targeted attacks on adversarially 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 handling multi-objective attack scenarios.

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 adversarially 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, table 2 indicates only a slight decrease in performance compared to the classifier without the defense mechanism.
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.

**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 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.

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.

<table>
<thead>
<tr>
<th>Method</th>
<th>ASR</th>
<th>$l_0$</th>
<th>$l_2$</th>
<th>SSIM</th>
<th>ASR</th>
<th>$l_0$</th>
<th>$l_2$</th>
<th>SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>SA-MOO*</td>
<td>99.20%</td>
<td>134.00</td>
<td>45.25</td>
<td>0.99</td>
<td>99.20%</td>
<td>133.83</td>
<td>45.47</td>
<td>0.99</td>
</tr>
<tr>
<td>Sparse-RS*</td>
<td>97.26%</td>
<td>146.72</td>
<td>65.89</td>
<td>0.98</td>
<td>97.60%</td>
<td>142.28</td>
<td>66.82</td>
<td>0.98</td>
</tr>
<tr>
<td>SAPF</td>
<td>94.10%</td>
<td>336.83</td>
<td>52.83</td>
<td>0.98</td>
<td>94.20%</td>
<td>322.60</td>
<td>51.07</td>
<td>0.98</td>
</tr>
<tr>
<td>Homotopy</td>
<td>89.72%</td>
<td>769.02</td>
<td>45.87</td>
<td>0.98</td>
<td>91.28%</td>
<td>749.01</td>
<td>68.92</td>
<td>0.98</td>
</tr>
<tr>
<td>GreedyFool</td>
<td>43.25%</td>
<td>1092.53</td>
<td>5.22</td>
<td>0.99</td>
<td>47.37%</td>
<td>993.14</td>
<td>5.02</td>
<td>0.99</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MobileNet</th>
<th>ASR</th>
<th>$l_0$</th>
<th>$l_2$</th>
<th>SSIM</th>
<th>ASR</th>
<th>$l_0$</th>
<th>$l_2$</th>
<th>SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>SA-MOO*</td>
<td>99.24%</td>
<td>135.44</td>
<td>40.51</td>
<td>0.99</td>
<td>97.58%</td>
<td>130.61</td>
<td>35.53</td>
<td>0.99</td>
</tr>
<tr>
<td>Sparse-RS*</td>
<td>97.69%</td>
<td>135.10</td>
<td>60.10</td>
<td>0.98</td>
<td>96.88%</td>
<td>134.39</td>
<td>59.12</td>
<td>0.98</td>
</tr>
<tr>
<td>SAPF</td>
<td>91.04%</td>
<td>3840.39</td>
<td>55.26</td>
<td>0.93</td>
<td>92.72%</td>
<td>4733.75</td>
<td>47.72</td>
<td>0.86</td>
</tr>
<tr>
<td>Homotopy</td>
<td>87.48%</td>
<td>802.42</td>
<td>42.03</td>
<td>0.95</td>
<td>86.42%</td>
<td>905.81</td>
<td>40.62</td>
<td>0.94</td>
</tr>
<tr>
<td>GreedyFool</td>
<td>38.82%</td>
<td>1127.0</td>
<td>5.92</td>
<td>0.99</td>
<td>41.18%</td>
<td>1721.32</td>
<td>5.95</td>
<td>0.99</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ResNet</th>
<th>ASR</th>
<th>$l_0$</th>
<th>$l_2$</th>
<th>SSIM</th>
<th>ASR</th>
<th>$l_0$</th>
<th>$l_2$</th>
<th>SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>SA-MOO*</td>
<td>98.33%</td>
<td>139.36</td>
<td>91.26</td>
<td>0.98</td>
<td>97.22%</td>
<td>142.82</td>
<td>95.72</td>
<td>0.98</td>
</tr>
<tr>
<td>Sparse-RS*</td>
<td>98.25%</td>
<td>139.79</td>
<td>91.59</td>
<td>0.98</td>
<td>96.92%</td>
<td>143.87</td>
<td>97.32</td>
<td>0.98</td>
</tr>
</tbody>
</table>

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.
References


Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. CoRR, 2017. 2


Yash Sharma and Pin-Yu Chen. Attacking the madry defense model with l1-based adversarial examples. CoRR, abs/1710.10733, 2017. 2


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. 2

