Towards Consistent and Efficient Decision-based Attacks

Henning Duwe
RWTH Aachen University, Germany
duwe@aim.rwth-aachen.de

Anna Münz
RWTH Aachen University, Germany
muenz@aim.rwth-aachen.de

Holger H. Hoos
RWTH Aachen University, Germany
Leiden University, The Netherlands
hh@aim.rwth-aachen.de

Abstract

Deep neural networks have revolutionised computer vision with their powerful capabilities. However, they remain vulnerable to adversarial attacks, which seek to find small perturbations to input images that mislead them into making incorrect predictions. Traditional attacks often rely on gradient information, which is typically unavailable to attackers in real-world scenarios. To address this, black-box adversarial attacks have been developed that do not depend on full access to the network. In this study, we focused on the most challenging black-box setting, decision-based attacks, where the target model only returns output labels. While previous approaches for this setting exist, they either perform inconsistently across different networks, or are resource-intensive concerning running time.

*To address this, we developed a variant of the Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES) with a selective query approach that enables optimisation for 80% more generations with the same query budget. Furthermore, we utilised state-of-the-art automated algorithm configuration techniques to optimise our attack and achieve a success rate of 100% using an initial search strategy. We compared our novel **D**ecision-based **A**ttack via **CMA-ES** (**DACES**) against various well-known attacks, demonstrating the best average performance in creating minimal adversarial perturbations across several network architectures for ImageNet and CIFAR-100. Notably, DACES is more than three times as fast with respect to running time than any other attack on ImageNet, making it particularly applicable to real-world settings.*

1. Introduction

Over the past decade, computer vision has made remarkable progress, enabling applications in industry, healthcare, and beyond. Despite this, deep neural networks remain vulnerable to adversarial attacks [11, 31]. Adversarial attacks create a perturbation δ that, when added to a correctly classified, original image $\mathbf{x}_{\text{orig}}$, causes $\mathbf{x}_{\text{orig}} + \delta$ to be misclassified. Studying these attacks not only helps us understand the weaknesses of neural networks, but also drives the development of new adversarial defences [4].

Adversarial attacks can be categorised into white-box and black-box attacks [21]. In white-box attacks, the attacker has full access to the architecture and parameters of a neural network, allowing them to compute gradients and find adversarial examples by moving the image in the direction that maximises the loss. Prominent white-box attacks include FGSM [11], PGD [24] and AutoAttack [5]. However, since external attackers rarely have full access to a neural network, research has started to consider black-box strategies, where the attacker can only access the predictions obtained from the classifier. The black-box setting is further divided into three categories [28]: transfer-based attacks [36], which use surrogate models to generate adversarial examples; score-based attacks [21, 26], where the attacker has access to prediction probabilities obtained from a neural network; and decision-based attacks [3, 7, 28, 34, 35], where only the output label is available.

Previous research used evolutionary algorithms because of their good performance in black-box optimisation [1, 7]. One popular method is the Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES), which has been extensively used for score-based attacks [7, 21, 26]. The decision-based setting, however,

Table 1. Average running time of each attack in seconds. HSJA is omitted as the only CPU-based method. Running DACES on CPU to compare it to HSJA reveals an advantage of 9.5% for DACES (4 332 compared to 4 743 seconds).

	DACES	CGBA	TA	(1+1)-CMA-ES
ImageNet	**25.0**	75.2	126.0	300.5
CIFAR-100	**23.6**	23.8	107.6	169.2

comes with multiple challenges: Score-based attacks can find adversarial examples by following the changing confidence scores of the classifier; in other words: slight modifications of the adversarial example will lead to different confidence scores – information which can be used to find a good region in the image space to attack. In the decision-based setting, however, one can only guide the search with discrete labels, which results in rugged loss landscapes. Even though the CMA-ES is well suited for these types of landscapes [12], to the best of our knowledge, only a simple variant of this algorithm, the (1+1)-CMA-ES has been successfully used for decision-based attacks [7]; this variant relies on a population size of only two for its search. In our work, for the first time, we investigated the effectiveness of the original CMA-ES in this setting. We used the diversity of the higher population size to create minimal adversarial perturbations and addressed the issue of the rugged loss landscapes encountered in the decision-based setting with an Initial Search Strategy (ISS). This ISS rapidly increases the variance of the covariance matrix to find the decision boundary of the neural network while keeping perturbations small.

An important consideration for adversarial attacks is the number of times the neural network is used for classification, the so-called number of "queries". CMA-ES naturally has a high query demand, which stems from the necessity of classifying all members of its population (rather than single instances) to assess the success of the attack. Here, one query is equivalent to running inference on one instance of the population. A high query demand is unfavourable, as deployers of machine learning models might notice heavy usage of their models and cut access after they have been extensively used. This means that for an attack to be usable in real-world scenarios, it has to be query-efficient. Thus, we implemented the first query strategy for this algorithm, allowing us to run the CMA-ES for 80% more iterations using the same query budget.

We also investigated the effectiveness of four different techniques that have been proposed to reduce the dimensionality of the problem. Searching for perturbations in a lower-dimensional subspace is necessary for most adversarial attacks, as the original image size is too large; for example, having an image with $3 \times 224 \times 224$ pixels results in a 150 528-dimensional problem. Based on our findings, we identified the best-performing dimensionality reduction technique for our attack. Finally, current research on decision-based attacks considered only query efficiency [7, 28, 35]. However, in real-world scenarios, time constraints might also apply, for example, when someone wants to find as many adversarial examples as possible in a limited time frame to determine the vulnerability of a specific neural network. Thus, we add running time as a dimension to the analysis of adversarial attacks. In summary, our contributions are:

- We introduce a **D**ecision-based **A**ttack via **C**MA-**ES** (**DACES**)[1] that leverages our newly developed query strategy, allowing it to run for 80% more generations within the same query budget. Our ablation studies show how this increase in iterations significantly reduces the magnitude of the perturbations needed to create adversarial examples.
- We designed a novel ISS and used state-of-the-art automated algorithm configuration techniques for our adapted CMA-ES. Remarkably, our attack demonstrated the best average performance in comparison to four state-of-the-art attacks over all ImageNet and CIFAR-100 networks.
- Our analysis of the running time of decision-based attacks in Table 1 reveals that DACES is more than three times as fast as any other attack on ImageNet and the fastest attack on CIFAR-100.
- Last, we show that DACES substantially contributes to the state of the art by analysing marginal contributions. In particular, an optimal ensemble attack of previous methods on ImageNet performs 55% worse compared to an ensemble with DACES.

The remainder of this article is structured as follows: In Section 2, we formally introduce the problem and the CMA-ES algorithm, followed by related work on adversarial attacks in Section 3. In Section 4, we outline our novel method, followed by Section 5, which presents the experiments conducted and the results obtained. Finally, in Section 6, we close with conclusions.

2. Background

In this section, we first introduce the problem definition of finding adversarial examples and then the CMA-ES.

2.1. Problem Definition

The task of finding adversarial examples can be defined as a simple optimisation problem [1]: Given an original, unperturbed image $\mathbf{x}_{\text{orig}}$ with its label y and

[1] Source code: https://github.com/srheduwe/DACES.

a classifier $f_\theta : \mathbb{R}^n \to \mathbb{R}^m$, where θ is the set of trained parameters, n is the size of the input features, and m is the number of classes, we try to find an adversarial example $\mathbf{x}_{\text{orig}} + \delta = \mathbf{x}$ such that $f_\theta(\mathbf{x}) \neq y$ and that minimises $||\mathbf{x} - \mathbf{x}_{\text{orig}}||_p$. Table 4 in Appendix A gives an overview of the notation used in this article. Regarding the distance metric, we follow the literature on decision-based attacks [23, 25, 27, 28] and focus on the Euclidean distance, *i.e.*, $p = 2$.

We considered the decision-based setting, where one has access to the input $\mathbf{x}_{\text{orig}}$, the target y, and the predicted label $f_\theta(\mathbf{x})$ but not to the internal information of the model such as the weights or to the output probabilities (or confidence scores). In contrast to targeted attacks, which seek to flip the target to a specific label, we focused on untargeted attacks, which allow the label to be flipped to any other class, as this is sufficient to demonstrate the vulnerability of neural networks [1].

2.2. CMA-ES

Evolutionary strategies are stochastic methods for real parameter optimisation of non-linear, non-convex functions [12] with many applications in engineering and machine learning [20]. These algorithms are based on the ideas of biological evolution, *i.e.*, variation and selection of the fittest instances of a population [20]. The CMA-ES is a special strategy that iteratively samples search points or solutions from a multivariate Gaussian distribution, evaluates their fitness, and selects the best ones, which in turn are used to fit the Gaussian of the next step or "generation". The equation for sampling solutions is given by [12]

$$\mathbf{x}_i^{(g+1)} \sim \mathcal{N}\left(\mathbf{m}^{(g)}, (\sigma^{(g)})^2 \mathbf{C}^{(g)}\right) \text{ for } i = 1, \ldots, \lambda, \quad (1)$$

where g is the generation number, $\mathbf{x}_i^{(g+1)} \in \mathbb{R}^n$ is the i-th search point from $g+1$, $\mathcal{N}(\cdot, \cdot)$ is a multivariate Gaussian distribution, $\mathbf{m}^{(g)} \in \mathbb{R}^n$ is the mean of the Gaussian for the g-th generation, $\sigma^{(g)} \in \mathbb{R}_{>0}$ is the standard deviation or step-size at g, $\mathbf{C}^{(g)}$ is the covariance matrix, and λ is the population size [12]. In each generation, CMA-ES updates the parameters of this Gaussian. The new mean is calculated as [12]:

$$\mathbf{m}^{(g+1)} = \mathbf{m}^{(g)} + c_{\text{m}} \cdot \sum_{i=1}^{\mu} w_i \cdot (\mathbf{x}_{i:\lambda}^{(g+1)} - \mathbf{m}^{(g)}), \quad (2)$$

where c_{m} is the learning rate for the mean, μ is the parent population size or, in other words, the number of selected points, $\mathbf{x}_{i:\lambda}^{(g+1)}$ is the i-th best sample at g+1, and w_i is the i-th weight coefficient for recombination [12]. The weights are determined by

$$w_i = \log((\lambda + 1)/2) - \log(i) \quad \text{for } i = 1, \ldots, \lambda, \quad (3)$$

which assigns positive weights for μ and negative weights otherwise (and only the positive weights are used for the update). In simple terms, at generation $g+1$, the mean is updated by adding the weighted difference between the new samples, $\mathbf{x}_{i:\lambda}^{(g+1)}$ and the mean at generation g, multiplied by a learning rate. While generally in each generation, 50% of the population (or, in other words, μ instances) are selected for fitting the new distribution, samples with a better fitness value are weighted more and are, thus, more important for the new distribution [12]. The part of the population that is most influential for adapting the Gaussian distribution is the effective population size μ_{eff}. It is determined by $\mu_{\text{eff}} = \frac{1}{\sum_{i=1}^{\mu} w_i^2}$ [12]. The effective population size will be used later as a part of the query strategy. For our experiments, we mostly set $\lambda = 23$, which results in an effective population size of $\mu_{\text{eff}} = 6$ accounting for roughly 88% of the weights.

Explaining the update of the covariance matrix is more involved and out of the scope of this article. The interested reader is referred to the tutorial and the references therein [12]. However, an important consideration for updating the covariance matrix is whether the full matrix is needed. Consider the example mentioned before, where we want to find an adversarial example for images of the size of $n = 3 \times 224 \times 224$, which resulted in a 150 528-dimensional problem. For this, the original CMA-ES would update and store a matrix of size $n \times n$, *i.e.*, with $150\,528 \cdot 150\,528$ covariances in each generation. Formally, the computation and storage complexity of the covariance matrix becomes at least $\mathcal{O}(n^2)$ [18]. A way to address this is to assume that the variables are independent, *i.e.*, to only adjust the diagonal parts and set the rest to 0, a strategy also adapted by [18]. This reduces both computation and storage complexity to $\mathcal{O}(n)$, allowing the use of CMA-ES for high-dimensional problems [29].

3. Related Work

In this section, we present relevant work on decision-based attacks, including those we compare ours against and finally, score-based attacks using CMA-ES.

3.1. Decision-based Attacks

Most decision-based attacks rely on gradient approximation and binary search [3, 27, 28]. In contrast, one of the earliest approaches, the Boundary Attack [2], samples random perturbations from a Gaussian distribution to find adversaries while minimising the l_2-norm, but is query inefficient compared to newer attacks [18].

The HopSkipJumpAttack (HSJA) [3], another early decision-based attack, initially identifies the decision boundary through binary search. By querying the model near the decision boundary, the authors approximate the gradient using Monte Carlo estimation, resulting in an attack vector normal to the decision boundary. They then iteratively improve the adversarial example by following the gradient.

Another approach, the Curvature-aware Geometric Black-box Attack (CGBA) [28], uses an initial boundary point from which it follows a random direction to find the nearest point that is still adversarial. At this new boundary point, solutions from a Gaussian distribution are sampled to estimate a vector that is orthogonal to the decision boundary, which is used to span a semicircular path between the initial image and the boundary point. On this semicircular path, a binary search is conducted between an adversarial and a non-adversarial example to find a closer adversary. The authors propose two versions of their algorithm, as the initial approach performs worse when "the curvature of the boundary is too high" [28].

The Triangle Attack (TA) [35] spans a triangle between the benign image, the adversarial example of the previous and the current step to find adversarial examples. The decision boundary is reached by adjusting the angles of the triangle. This simple attack was shown to be query efficient and to perform well for low query budgets [35].

Dong *et al.* proposed an attack with the (1+1)-CMA-ES [7]. In this variant, in every generation, only a single offspring is generated from the parent by adding random noise; from these two solutions, the one with better fitness is then used for the next iteration. This attack has been mainly evaluated on face recognition models, which are especially vulnerable to attacks [7].

3.2. Score-based Attacks with CMA-ES

In the less difficult score-based setting, further approaches using CMA-ES have been proposed. For instance, Qiu *et al.* [26] compared CMA-ES with two other evolutionary algorithms and found that the CMA-ES outperforms them. Their research focused on the number of queries required to create adversarial examples below certain l_∞-norm-thresholds. A method closely related to our work is the Subspace Activation Evolutionary Strategy (SA-ES) [21], which is based on the idea of reducing the search space for the evolutionary algorithm by binarising the red channel of the smoothed images to identify an "active subspace". In the binarised red channel, the largest patch of 1s is found, enforcing a minimum patch size of 8×8, and this subspace is then used to attack the image over all three channels. With this technique, Li *et al.* reduced the dimensionality of the images by around 43%.

Despite the success of CMA-ES in the score-based setting and one of its variants, (1+1)-CMA-ES, in decision-based attacks, the performance of CMA-ES with larger population sizes for the second setting has not yet been investigated. This is likely due to the inherently high query demand of this algorithm; every generation usually requires classification of each instance in the population, which makes this algorithm not competitive. To address this, we developed the first query strategy for CMA-ES-based adversarial attacks and demonstrate how this significantly reduces the inference costs of our attack. By doing so, we address multiple limitations of previous decision-based attacks. These often try to exploit geometric properties of the decision boundaries to find adversarial examples [27, 28], which hinders performance when the geometric properties are different from those anticipated. Then, there are approaches that are not optimised for modern GPUs [3] and, thus, suffer from high running times. The (1+1)-CMA-ES uses a general-purpose algorithm to find adversarial examples, but also incurs high running times, as only one new image is sampled in each iteration [7]. We address these limitations by using the original CMA-ES algorithm to create an adversarial attack that performs well on different network architectures with differing decision boundaries, but also excels in running time.

4. DACES

In this section, we formally describe our adaptations to the CMA-ES algorithm, *i.e.*, the ISS and querying strategy that make it more suitable for searching adversaries. We then describe the dimensionality reduction methods used to scale down the optimisation problem.

We define a simple fitness function F for our algorithm [1]. The generated images must be classified differently from $\mathbf{x}_{\text{orig}}$ while being as close as possible to it concerning the l_2-norm. Thus, we minimise

$$F(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}_{\text{orig}}\|_2 + l(\mathbf{x}), \tag{4}$$

where $\|\mathbf{x} - \mathbf{x}_{\text{orig}}\|_2$ is the Euclidean distance between the adversarial example $\mathbf{x}$ and the original image $\mathbf{x}_{\text{orig}}$, and $l(\mathbf{x})$ is a term that promotes $\mathbf{x}$ being adversarial:

$$l(x) = \begin{cases} 0 & \text{if } f_\theta(\mathbf{x}) \neq y \\ m & \text{otherwise,} \end{cases}, \tag{5}$$

where m is a constant that is usually set to high values, *e.g.*, $m = \infty$ [7]. To optimise the target function, we adapted the well-known CMA-ES algorithm and used the values of the target function as fitness

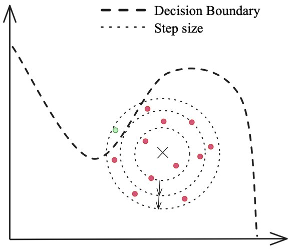


Figure 1. Illustration of the Initial Search Strategy for the step-size σ. The cross represents the input image $\mathbf{x}_{\text{orig}}$, the red dots samples with label $f_\theta(\mathbf{x}) = y$, and the green dot a sample with label $f_\theta(\mathbf{x}) \neq y$. σ is increased quickly to find an adversarial example within the first generations.

values. We treated our problem as separable, *i.e.*, we only adjusted the diagonal parts of the covariance matrix and set the rest to 0, effectively assuming that the variables are independent. In the context of decision-based attacks on image classifiers, each search point $\mathbf{x}_i^{(g+1)}$ corresponds to one possible perturbation δ in the image space. In other words, in each generation, CMA-ES proposes λ perturbations for the adversarial attack. While this distribution is interesting in itself, we focused on the single best point of the distribution as a result of our adversarial attack. To avoid a tedious, manual search for good parameters for our attack, we employed automated algorithm configuration; we provide details for this in Appendix B and explain our adaptations to the CMA-ES in the following sections.

4.1. Initial Search Strategy

A crucial aspect is the initialisation of the CMA-ES. Recall that each instance in the population represents a perturbation δ that is added to $\mathbf{x}_{\text{orig}}$, *i.e.*, we create adversarial examples with $\mathbf{x}_{\text{orig}} + \delta = \mathbf{x}$. Ultimately, we aim for small perturbations that lead to misclassification. In Figure 1, this region is located around the green dot. Thus, it intuitively makes sense to initialise the mean of the search distribution with $\mathbf{m}^{(0)} = 0$, effectively sampling adversarial examples around $\mathbf{x}_{\text{orig}}$.

However, this often results in no adversarial instances in the first generation, especially when $\mathbf{x}_{\text{orig}}$ is far from the decision boundary. This issue is less relevant in the score-based setting, where confidence scores guide the population towards the adversarial region [18, 26]. In the decision-based setting, only the label is available, necessitating a penalty term (in our case, $l(\mathbf{x})$). When all samples are benign (or adversarial), $l(\mathbf{x})$ becomes ineffective, as the fitness of every sample is added with the same term. Therefore, it is unlikely that the CMA-ES will find any adversarial examples later on. One can also initialise the mean randomly, starting with a large perturbation likely to cause misclassification from the first generation, as suggested by Dong *et al.* [7]. However, this necessitates more generations to reduce the perturbation size.

Our approach aims to make the warm start of initialising with $\mathbf{m}^{(0)} = 0$ usable for decision-based attacks. We achieve this by multiplying the step-size (or standard deviation) σ each generation with a configurable parameter $\exp(e_\sigma)$, while fixing $\mathbf{m}^{(g)} = 0$, as illustrated in Figure 1. This corresponds to lines 9-10 in Algorithm 1. Only after the algorithm finds adversarial examples, it returns to the optimisation process. With our population size $\lambda = 23$ and $t_{\text{adv}} = 0.95$, a single sample needs to be adversarial to return to the normal optimisation process. The effectiveness of this strategy can be explained by the properties of the loss landscape defined by the fitness function $F(\mathbf{x})$. For correctly classified images, the term $l(\mathbf{x})$ is very large compared to the Euclidean distance $\|\mathbf{x} - \mathbf{x}_{\text{orig}}\|_2$. To avoid convergence to local minima, DACES prioritises exploration through the initial search. Once the first adversary is found, the fitness function decreases sharply, DACES stops the initial search and transitions towards exploitation. This procedure is detailed in lines 11-14 of Algorithm 1.

Algorithm 1 DACES[2]

```
1:  Input: Image $\mathbf{x}_{\text{orig}}$, label $y$, neural network $f_\theta$,
    number of queries $Q$, ISS threshold $t_{\text{adv}}$, effective
    population size $\mu_{\text{eff}}$
2:  Output: Perturbation $\delta$
3:  Initialisation: $\mathbf{m}^{(0)} = 0$, $g = 0$, dimensionality $n$
4:  while $q < Q$ do
5:      for $i \leftarrow 0$ to $\lambda$ do
6:          $\mathbf{x}_i^{(g+1)} \sim \mathcal{N}\left(\mathbf{m}^{(g)}, (\sigma^{(g)})^2\mathbf{C}^{(g)}\right)$
7:      $F^{(g+1)} \leftarrow \textsc{Fitness}(\mathbf{x}^{(g+1)}, \mathbf{x}_{\text{orig}}, y, f_\theta, \mu_{\text{eff}})$
                                                    ▷ Call Algorithm 2
8:      if $\frac{1}{\lambda} \cdot \sum_{i=1}^{\lambda} \mathbf{1}(f_\theta(\mathbf{x}_i^{(g+1)}) = y) \geq t_{\text{adv}}$ then
            ▷ Update $\sigma$ based on % of adversarial examples
9:          $\sigma^{(g+1)} = \exp(e_\sigma)\,\sigma^{(g)}$ ▷ Manually set a high $\sigma$
10:     else
11:         $\sigma^{(g+1)} \leftarrow \text{UPDATE\_SIGMA}(\sigma^{(g)})$
                                    ▷ Otherwise, update $\sigma$ normally
12:         $w_i = \log((\lambda+1)/2) - \log(i)$   for $i = 1, \ldots, \lambda$
                                              ▷ Calculate weights
13:         $\mathbf{m}^{(g+1)} = \mathbf{m}^{(g)} + c_{\text{m}} \cdot \sum_{i=1}^{\mu} w_i(\mathbf{x}_{i:\lambda}^{(g+1)} - \mathbf{m}^{(g)})$
                                              ▷ Update the mean
```

[2] Table 4 gives an overview of the notation used.

4.2. Query Strategy

To find out whether the instances of the population have crossed the decision boundary or not, they need to be classified. Classifying the entire population is effective, but leads to a high number of queries. To be precise, for 1000 generations with a population size of 50, one would query the model 50 000 times. A more sophisticated approach is therefore required.

To reduce the number of queries, we divided the optimisation process into three stages, as illustrated in Figure 3 and formalised in Algorithm 2. In each stage, an increasing proportion of the population is queried. In stage 1 (lines 3-4 of Algorithm 2), the algorithm tests whether a valid, *i.e.*, adversarial, solution already exists, by querying the single instance closest to the original image (in terms of the l_2-norm). The rationale behind this is that as long as the population is far from the decision boundary, the query budget should be conserved. After the single closest instance has crossed the decision boundary once, *i.e.*, it has been classified identically to the original image, more of the query budget is used. In the pseudocode, this is indicated with a Boolean indicator C_{ben}, which states whether the closest sample has been classified as benign before. In the second stage (lines 5-7), the effective population size μ_{eff} is queried; for our problem, this is usually around 30% of the total population. Finally, as the population converges and gets closer to the decision boundary, the entire population is queried. This third stage (lines 8-9) is entered as soon as 75% of μ_{eff} has crossed the decision boundary. Intuitively, the query strategy works because it preserves query budget for the more critical phases of the optimisation, where more information (in form of classifications) is needed to obtain a better result.

Note that we allow transitioning back from stage 3 to stage 2, but not from stage 2 or 3 to stage 1. The rationale behind this is that by only querying a single instance, we implicitly assume that the remaining population has not crossed the decision boundary yet; otherwise, we must have added $l(\mathbf{x})$ to their fitness values. This assumption holds only when the population is still far from the decision boundary.

Algorithm 2 FITNESS VALUES

```
1: Input: Population $\mathbf{x}^{(g+1)}$, image $\mathbf{x}_{orig}$, label $y$, neural network $f_\theta$, effective population size $\mu_{eff}$, boolean indicator $C_{ben}$
2: Output: Fitness values $F^{(g+1)}$
3: if $f_\theta(\mathbf{x}_{1:\lambda}^{(g+1)}) \neq y$ and not $C_{\text{ben}}$ then
4:     $F_{i:\lambda}^{(g+1)} = ||\mathbf{x}_{i:\lambda}^{(g+1)} - x||_2$    for $i = 1, \dots, \lambda$
5: else if $\frac{1}{\mu_{\text{eff}}} \cdot \sum_{i=1}^{\mu_{\text{eff}}} \mathbf{1}(f_\theta(\mathbf{x}_i^{(g+1)}) = y) < 0.75$ then
6:     $F_{i:\lambda}^{(g+1)} = ||\mathbf{x}_{i:\lambda}^{(g+1)} - x||_2 + l(\mathbf{x}_{i:\lambda}^{(g+1)})$ for $i = 1, \dots, \mu_{\text{eff}}$    ▷ Penalty for benign examples
7:     $F_{i:\lambda}^{(g+1)} = ||\mathbf{x}_{i:\lambda}^{(g+1)} - x||_2$    for $i = \mu_{\text{eff}} + 1, \dots, \lambda$    ▷ No penalty for rest
8: else
9:     $F_{i:\lambda}^{(g+1)} = ||\mathbf{x}_{i:\lambda}^{(g+1)} - x||_2 + l(\mathbf{x}_{i:\lambda}^{(g+1)})$ for $i = 1, \dots, \lambda$    ▷ Classify whole population
```

4.3. Dimensionality Reduction

Finding adversarial examples in the original image space is not feasible for datasets such as ImageNet in the decision-based setting due to the high dimensionality. Therefore, researchers have used different techniques to find lower-dimensional subspaces to attack and thus render adversarial example generation feasible in such cases. We investigated four different dimensionality reduction techniques. However, two of these inherently limit the number of pixels that can be attacked. Since no method we compare against considered this type of limitation, we cover these only in the supplementary material (Appendix D).

For the remainder of our study, we focus on the well-known Bilinear Interpolation (BI) and Nearest Neighbour Interpolation (NNI). BI is the two-dimensional generalisation of linear interpolation and is used, for example, in the (1+1)-CMA-ES-based attack of Dong *et al.* [7]. In essence, linear interpolation is first applied along the x-axis to find intermediate values. This process is then repeated along the y-axis, using the intermediate values to obtain the final interpolated values. For details, we refer to Merchant *et al.* [37]. A perturbation created in a 3 × 14 × 14 space and then upscaled to the full image size of 3 × 224 × 224 with BI can be seen in Appendix D, Figure 4(a).

NNI was used by Qiu *et al.* [26] in their score-based attack that uses CMA-ES. In simple terms, NNI scales a picture by assigning the colour of the nearest pixel to the new pixels created during the scaling process, which can result in a blocky or pixelated appearance. See Appendix D.4 for details. A perturbation created in a 3 × 16 × 16 space and then upscaled to the full image size can be seen in Appendix D, Figure 4(b). The performance of our decision-based attack using both NNI and BI will be compared against each other.

5. Experiments

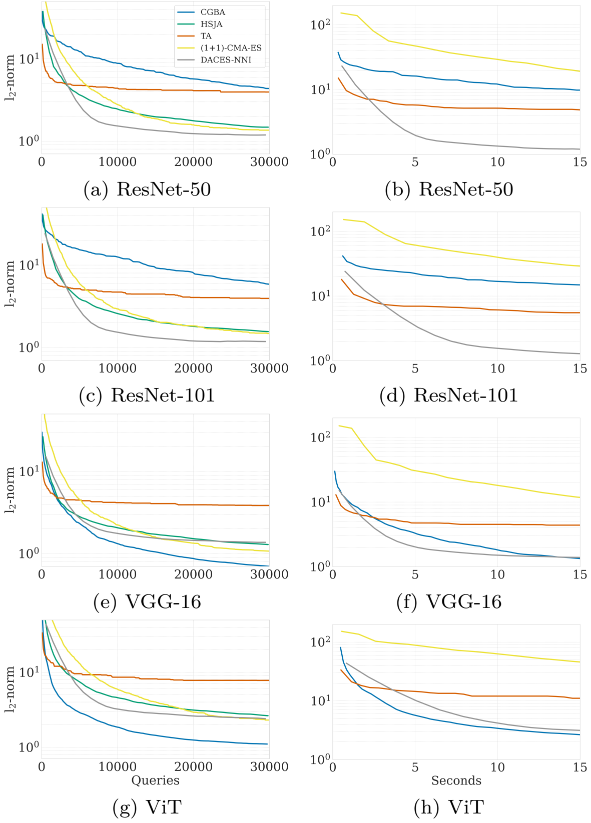
Here, we describe the experiments we conducted in our research that demonstrate the efficacy of DACES and its components, namely the query strategy and the ISS. We compared DACES using BI (DACES-BI) and NNI (DACES-NNI) with HSJA [3], TA [35], (1+1)-CMA-

ES [8], and CGBA [28].[3] We used the EvoTorch library [33] as the basis for our work. Each attack is evaluated on 200 correctly classified images which were selected uniformly at random, without replacement, from the testing sets of ImageNet and CIFAR-100. All experiments designed for GPUs, namely CGBA, TA, (1+1)-CMA-ES and our attack, were run on an NVIDIA H100 GPU with 80 GB of RAM, and HSJA was run on an AMD EPYC 7543 2.8GHz with 1 TB of RAM, as HSJA cannot be trivially transferred to GPU. Due to the wide availability of GPUs, virtually all newer attacks take advantage of them. However, it is important to note that DACES is still the fastest in terms of running time, independent of the hardware (*i.e.* GPU or CPU).

5.1. Evaluation Setup

We used ImageNet [6] and CIFAR-100 [15] for evaluation. ImageNet is a large image classification dataset containing 1 281 167 training, 50 000 validation, and 100 000 test images over 1 000 classes. CIFAR-100 [15] is a small dataset with 50 000 training and 10 000 test images of $3 \times 32 \times 32$ over 100 classes. Due to this size, we did not use any downscaling techniques when attacking CIFAR-100 and refer to DACES in this context (rather than differentiating DACES-BI and -NNI).

Following Reza *et al.* [28], we evaluated our attack on four different neural network classifiers for ImageNet: ResNet-50, ResNet-101 [13], VGG-16 [30] and ViT [9]. For these networks, images were rescaled to $3 \times 224 \times 224$. Then, we use three adversarially-trained models from MadryLab,[4] that take input sizes of $3 \times 299 \times 299$, which we will abbreviate with Robust RN-50, Robust RN-101 and Robust VGG-16. For CIFAR-100, we used the same network architectures trained on that dataset, except that we replaced ResNet-101 by ResNet-34, due to the lower complexity of CIFAR-100. We used two metrics to evaluate all attacks [28]: the median l_2-norm perturbation (in relation to the number of queries or running time) and the Area Under the Curve (AUC), which is defined as follows: Let q_i be the number of queries at step i, and let δ_i be the perturbation vector. Then the AUC is defined as $\text{AUC} = \sum_{i=1}^{N} q_i \cdot \|\delta_i\|_2$ over N steps of the attack. The AUC can be interpreted as the cumulative perturbation cost over the course of the attack, which summarises the performance across all query budgets. For comparison with DACES, we used the original implementations and parameters of HSJA [3], TA [35], CGBA [28] and (1+1)-CMA-ES [8]. We ran each attack for 30 000 queries. Due to the stochasticity of our method, we averaged performances over ten runs using different seeds.



(a) ResNet-50 (b) ResNet-50

(c) ResNet-101 (d) ResNet-101

(e) VGG-16 (f) VGG-16

(g) ViT (h) ViT

Figure 2. Median distance versus number of model queries/time over 200 images from the ImageNet testing set (lower is better). For better readability, we concentrate on the better-performing DACES-NNI and neglect DACES-BI. DACES-NNI shows a consistently good performance over the ResNets, the VGG-16 and ViT architectures.

5.2. Results

Table 2 and Table 3 display the results of all attacks on ImageNet and CIFAR-100 by reporting the AUC. Additionally, Table 7 and Table 10 show the performance of the attacks at 1 000, 10 000, and 30 000 queries. Generally, we observed DACES-BI and DACES-NNI to perform similarly, with an advantage for the NNI variant, which we therefore decided to focus on (see Appendix I.2). Across all ImageNet networks, DACES-NNI was observed to obtain the best average AUC, followed by DACES-BI, HSJA, TA, (1+1)-CMA-ES and last CGBA. On CIFAR-100, HSJA performs best, followed by CGBA and DACES. We discuss possible explanations for the strengths and weaknesses of DACES and conduct statistical analysis in Appendix H.

[3] We used CGBA_H only, as CGBA failed in most cases.

[4] https://github.com/MadryLab/robustness

Table 2. AUC over 200 images (per network) from the ImageNet testing set (lower is better).

Attack	ResNet-50	ResNet-101	VGG-16	ViT	Robust RN-50	Robust RN-101	Robust VGG-16	Average
DACES-BI	122 402	128 560	118 100	222 011	221 207	215 883	173 201	171 623
DACES-NNI	**107 729**	**114 174**	90 308	214 425	**170 432**	**185 084**	142 120	**146 325**
HSJA	122 970	129 996	85 002	218 035	244 118	363 201	139 026	186 050
CGBA	377 780	435 364	**78 872**	**103 177**	286 838	714 723	**128 073**	303 547
TA	191 133	203 052	161 763	380 992	305 901	334 938	222 726	257 215
(1+1)-CMA-ES	213 668	217 533	161 167	379 982	331 156	292 567	190 376	255 207

Table 3. AUC over 200 images (per network) from CIFAR-100 testing set (lower is better).

Attack	ResNet-50	ResNet-34	VGG-16	ViT	Average
DACES	21 241	20 484	20 212	101 338	40 819
HSJA	**5 535**	**5 470**	**6 338**	127 058	36 100
CGBA	25 140	26 766	24 065	**52 114**	**32 021**
TA	67 023	56 819	60 979	156 501	85 331
(1+1)-CMA-ES	17 672	17 950	19 719	307 931	90 818

Figure 2 displays the results on ImageNet only. For each attack, the median l_2-norm over 200 images is shown on a logarithmic scale over the mean number of queries or time. On the ResNet-50 and ResNet-101 models, the two variants of DACES perform best on all query budgets above 3 000, up to which TA performs best. On VGG-16, CGBA dominated from 2 000 queries onwards and on ViT across all query budgets. On ViT, DACES performed second best.

Finally, we compared the running times of the different attacks, which, in our experiments, were measured in terms of wall clock time. We added this to the analysis of the attacks because in real-world scenarios, not only query but also time limits might apply. HSJA has been omitted as it was designed for running on CPUs and is not trivially convertible to make use of GPUs. To compare against HSJA, we ran our attack for the ResNet-50 on CPUs, which resulted in an average running time per image of 4 332 seconds compared to 4 743 for HSJA. The main running time comparison can be found in Table 1. DACES-NNI is the fastest attack on ImageNet, where it was observed to be 3 times faster on average than the next quickest attack, CGBA. These results show that DACES is specifically suitable for attacking large ImageNet networks in a setting where not only the query budget but also the time is constrained. Another piece of evidence for this is the right side of Figure 2, where it can be seen that DACES performed even better than in the query-restricted setting, clearly emerging as the best or second-best algorithm over all networks in performance with respect to time.

We provide further results in Appendix F and G. We also performed ablation studies to investigate the effectiveness of the ISS, Automated Algorithm Configuration (AAC) and our query strategy (Appendix I).

The performance on the Robust RN-50, Robust RN-101 and Robust VGG-16 is similar to their normally trained counterparts with DACES-NNI dominating on the former two and performing second best on the latter with respect to queries (Table 2), but performing superior over nearly all time budgets.

5.3. Marginal Contribution of DACES

While the previous metrics are useful for comparing the performance of our attack to other methods, they provide no information about the marginal contribution of DACES to the state of the art, in that they do not indicate whether a combination of multiple other attacks could achieve the same performance as ours. To do so, we create a so-called Virtual Best Attack (VBA). The VBA is a hypothetical attack that optimally combines a set of attacks by picking the best one for each image; in other words, it is the minimum perturbation achieved by all attacks on an image. For ImageNet, we find that the VBA achieves an average l_2-norm of 1.44. By excluding DACES from this VBA (*i.e.* creating an ensemble attack of HSJA, CGBA, TA and (1+1)-CMA-ES only), this grows to 2.26, a plus of 55%. More details can be found in Appendix E.

6. Conclusion

In this study, we introduced DACES, a decision-based attack for image classification utilising CMA-ES. Our comparison with HSJA, TA, CGBA and (1+1)-CMA-ES revealed that DACES achieves the best average performance on various neural networks for ImageNet and CIFAR-100. We achieved this by implementing a novel initial search strategy and query strategy. The former warm-starts the search for adversarial examples, thus boosting the performance for low query budgets, and the latter reduces the number of queries used in each generation, resulting in 80% more iterations.

Our findings also emphasise the importance of running time as a quality metric for adversarial attacks, which has been largely overlooked in previous studies. DACES fully utilises modern GPUs, making it 3 times faster than the next quickest attack on ImageNet.

Acknowledgments

The authors thank Roman Kalkreuth, Konstantin Kaulen and Hadar Shavit for their useful feedback on this paper. This work has been partly funded by the Alexander von Humboldt Foundation through an Alexander von Humboldt Professorship awarded to Holger Hoos.

References

[1] Antony Bartlett, Cynthia C.S. Liem, and Annibale Panichella. Multi-objective differential evolution in the generation of adversarial examples. *Science of Computer Programming*, 238:103169, 2024. 1, 2, 3, 4

[2] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *International Conference on Learning Representations (ICLR)*, pages 1–10, 2018. 3

[3] Jianbo Chen, Michael I. Jordan, and Martin J. Wainwright. Hopskipjumpattack: A query-efficient decision-based attack. In *IEEE Symposium on Security and Privacy*, pages 1277–1294, 2020. 1, 3, 4, 6, 7, 18, 20

[4] 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 *ACM Workshop on Artificial Intelligence and Security (AISec)*, page 15–26, 2017. 1

[5] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International Conference on Machine Learning (ICML)*, pages 2206–2216, 2020. 1

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009. 7

[7] Yinpeng Dong, Hang Su, Baoyuan Wu, Zhifeng Li, Wei Liu, Tong Zhang, and Jun Zhu. Efficient decision-based black-box adversarial attacks on face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7714–7722, 2019. 1, 2, 4, 5, 6, 20

[8] Yinpeng Dong, Qi-An Fu, Xiao Yang, Tianyu Pang, Hang Su, Zihao Xiao, and Jun Zhu. Benchmarking adversarial robustness on image classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 318–328, 2020. 7

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, pages 1–9, 2021. 7

[10] Stefan Falkner, Aaron Klein, and Frank Hutter. BOHB: Robust and efficient hyperparameter optimization at scale. In *International Conference on Machine Learning (ICML)*, pages 1437–1446, 2018. 11

[11] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, pages 1–10, 2015. 1

[12] Nikolaus Hansen. The CMA evolution strategy: A tutorial. *(Preprint) arXiv:1604.00772*, pages 1–24, 2016. 2, 3

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 7

[14] Pascal Kerschke, Holger H. Hoos, Frank Neumann, and Heike Trautmann. Automated algorithm selection: Survey and perspectives. *Evolutionary Computation*, 27(1):3–45, 2019. 14

[15] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-100 (Canadian institute for advanced research), 2009. 7

[16] Fei-Fei Li, Marco Andreeto, Marc'Aurelio Ranzato, and Pietro Perona. Caltech 101, 2022. 20

[17] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, page 6391–6401, 2018. 17

[18] Huichen Li, Linyi Li, Xiaojun Xu, Xiaolu Zhang, Shuang Yang, and Bo Li. Nonlinear projection based gradient estimation for query efficient blackbox attacks. In *International Conference on Artificial Intelligence and Statistics*, pages 3142–3150, 2021. 3, 5

[19] Lisha Li, Kevin G. Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: Bandit-based configuration evaluation for hyperparameter optimization. In *International Conference on Learning Representations (ICLR)*, pages 1–8, 2017. 11

[20] Zhenhua Li, Xi Lin, Qingfu Zhang, and Hai-Lin Liu. Evolution strategies for continuous optimization: A survey of the state-of-the-art. *Swarm and Evolutionary Compututation*, 56:100694, 2020. 3

[21] Zhenhua Li, Huilin Cheng, Xinye Cai, Jun Zhao, and Qingfu Zhang. SA-ES: Subspace activation evolution strategy for black-box adversarial attacks. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 7(3):780–790, 2023. 1, 4, 13

[22] Marius Lindauer, Katharina Eggensperger, Matthias Feurer, André Biedenkapp, Difan Deng, Carolin Benjamins, Tim Ruhkopf, René Sass, and Frank Hutter. Smac3: A versatile bayesian optimization package for hyperparameter optimization. *Journal of Machine Learning Research (JMLR)*, 23(54):1–9, 2022. 11

[23] Chen Ma, Xiangyu Guo, Li Chen, Jun-Hai Yong, and Yisen Wang. Finding optimal tangent points for reducing distortions of hard-label attacks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 19288–19300, 2021. 3

[24] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, pages 1–15, 2018. 1

[25] Thibault Maho, Teddy Furon, and Erwan Le Merrer. Surfree: A fast surrogate-free black-box attack. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10425–10434, 2021. 3

[26] Hao Qiu, Leonardo Lucio Custode, and Giovanni Iacca. Black-box adversarial attacks using evolution strategies. In *Genetic and Evolutionary Computation Conference (GECCO)*, page 1827–1833, 2021. 1, 4, 5, 6, 14

[27] Ali Rahmati, Seyed-Mohsen Moosavi-Dezfooli, Pascal Frossard, and Huaiyu Dai. Geoda: A geometric framework for black-box adversarial attacks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8443–8452, 2020. 3, 4

[28] Md Farhamdur Reza, Ali Rahmati, Tianfu Wu, and Huaiyu Dai. CGBA: Curvature-aware geometric black-box attack. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 124–133, 2023. 1, 2, 3, 4, 7, 17, 20

[29] Raymond Ros and Nikolaus Hansen. A simple modification in CMA-ES achieving linear time and space complexity. In *Parallel Problem Solving from Nature - (PPSN)*, pages 296–305, 2008. 3

[30] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, pages 1–8, 2015. 7

[31] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, pages 1–10, 2014. 1

[32] 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*, 4(2):268–281, 2023. 13

[33] Nihat Engin Toklu, Timothy Atkinson, Vojtěch Micka, Paweł Liskowski, and Rupesh Kumar Srivastava. EvoTorch: Scalable evolutionary computation in Python. *(Preprint) arXiv:2302.12600*, pages 1–19, 2023. 7, 18

[34] Feiyang Wang, Xingquan Zuo, Hai Huang, and Gang Chen. TtBA: Two-third bridge approach for decision-based adversarial attack. In *International Conference on Machine Learning (ICML)*, 2025. 1

[35] Xiaosen Wang, Zeliang Zhang, Kangheng Tong, Dihong Gong, Kun He, Zhifeng Li, and Wei Liu. Triangle attack: A query-efficient decision-based adversarial attack. In *European Conference on Computer Vision (ECCV)*, pages 156–174, 2022. 1, 2, 4, 6, 7, 20

[36] Dongxian Wu, Yisen Wang, Shu-Tao Xia, James Bailey, and Xingjun Ma. Skip connections matter: On the transferability of adversarial examples generated with resnets. In *International Conference on Learning Representations (ICLR)*, pages 1–10, 2020. 1

[37] Qiang Wu, Fatima A. Merchant, and Kenneth R. Castleman. *Microscope Image Processing*. Elsevier, Academic Press, 2008. 6

[38] Jie Zhang, Christian Schlarmann, Kristina Nikolić, Nicholas Carlini, Francesco Croce, Matthias Hein, and Florian Tramèr. Evaluating the robustness of the "ensemble everything everywhere" defense. *(Preprint) arXiv:2411.14834*, 2025. 17