A Model-Based Derivative-Free Approach to Black-Box Adversarial Examples: BOBYQA

Ughi Giuseppe 1 Abrol Vinayak 1 Tanner Jared 1

Abstract
We demonstrate that model-based derivative free optimisation algorithms can generate adversarial targeted misclassification of deep networks using fewer network queries than non-model-based methods. Specifically, we consider the black-box setting, and show that the number of networks queries is less impacted by making the task more challenging either through reducing the allowed $\ell_\infty$ perturbation energy or training the network with defences against adversarial misclassification. We illustrate this by contrasting the BOBYQA algorithm (Powell, 2009) with the state-of-the-art model-free adversarial targeted misclassification approaches based on genetic (Alzantot et al., 2019), combinatorial (Moon et al., 2019), and direct-search (Andriushchenko et al., 2019) algorithms. We observe that for high $\ell_\infty$ energy perturbations on networks, the aforementioned simpler model-free methods require the fewest queries. In contrast, the proposed BOBYQA based method achieves state-of-the-art results when the perturbation energy decreases, or if the network is trained against adversarial perturbations.

1. Introduction
Deep neural networks (NNs) achieve state-of-the-art performance on a growing number of applications such as acoustic modelling, image classification, and fake news detection (Hinton et al., 2012; He et al., 2015; Monti et al., 2019) to name but a few. Alongside their growing application, there is a literature on the robustness of deep nets which shows that it is often possible to generate images with subtle perturbations, referred to as adversarial examples (Szegedy et al., 2014; Goodfellow et al., 2015), to the input of a network resulting in its performance being severely degraded; for example, see (Dalvi et al., 2004; Kurakin et al., 2017; Sitawarin et al., 2018; Eykholt et al., 2018; Yuan et al., 2019) concerning the use-case of self driving cars.

Methods to generate these adversarial examples are classified according to two main criteria (Yuan et al., 2019).

Adversarial Specificity establishes what the aim of the adversary is. In non-targeted attacks, the method perturbs the image in such a way that it is misclassified into any different category than the original one. While in targeted settings, the adversary specifies a category into which an image has to be misclassified.

Adversary’s Knowledge defines the amount of information available to the adversary. In White-box settings the adversary has complete knowledge of the network architecture and weights, while in the Black-box setting the adversary is only able to obtain the pre-classification output vector for a limited number of inputs. The White-box setting allows for the use of gradients of a missclassification objective to efficiently compute the adversarial example (Goodfellow et al., 2015; Carlini & Wagner, 2017; Chen et al., 2018), while the same optimization formulation of the Black-box setting requires use of a derivative free approach (Narodytska & Kasiviswanathan, 2017; Chen et al., 2017; Ilyas et al., 2018; Alzantot et al., 2019).

The generation of black-box targeted adversarial examples for deep NNs has been extensively studied in a setting initially proposed by (Chen et al., 2017) where:

- the adversarial example is found by solving an optimisation problem designed to change the original classification of a specific input to a specific alternative.
- the perturbation, which causes the network to change the classification, has entries bounded in magnitude by a specified infinity norm (maximum entry magnitude).
- the number of queries to the NN needed to generate the adversarial example should be as small as possible.
when considering Norm and Adv respectively. Here the (Moon et al., 2019) is a direct-search method that explores (Norm) and with the distillation defence (Adv) (Papernot and more robust networks. Model-based DFO is a well developed area, and we expect further improvements are possible through a more extensive investigation of these approaches.

2. Adversarial Examples Formulated as an Optimisation Problem

Consider a classification operator $F : \mathcal{X} \rightarrow \mathcal{C}$ from input space $\mathcal{X}$ to output space $\mathcal{C}$ of classes. A targeted adversarial perturbation $\eta$ to an input $X \in \mathcal{X}$ has the property that it changes the classification to a specified target class $t$, i.e. $F(X) = c$ and $F(X + \eta) = t \neq c$. Herein we follow the formulation by (Alzantot et al., 2019). Given: an image X, a maximum energy budget $\varepsilon_{\infty}$, and a suitable loss function $\mathcal{L}$, then the task of computing the adversarial perturbation $\eta$ can be cast as an optimisation problem such as

$$\min_{\eta} \mathcal{L}(X, \eta) \quad \text{s.t.} \quad \|\eta\|_{\infty} \leq \varepsilon_{\infty} \quad (1)$$

where the final two inequality constraints are due to the input entries being restricted to $[l, u]^n$. Denoting the pre-classification output vector by $f(X)$, i.e. $F(X) = \arg \max_{c} f(X)$, then the misclassification of X to target label $t$ is achieved by $\eta$ if $f(X + \eta) \geq \max_{j \neq t} f(X + \eta)_j$. In (Carlini & Wagner, 2017; Chen et al., 2017; Alzantot et al., 2019) they determined

$$\mathcal{L}(X, \eta) = \log (\Sigma_{j \neq t} f(X + \eta)_j) - \log (f(X + \eta)_t) \quad (2)$$

to be the most effective loss function for computing $\eta$ in (1), and we also employ this choice throughout our experiments.
3. Derivative Free Optimisation for Adversarial Examples

Derivative Free Optimisation is a well-developed field with numerous types of algorithms, see [Conn et al., 2009] and [Larson et al., 2019] for reviews on DFO principles and algorithms. Examples of classes of such methods include: direct search methods such as simplex, model-based methods, hybrid methods such as finite differences or implicit filtering, as well as randomized variants of the aforementioned and methods specific to convex or noisy objectives. The optimization formulation in Section 2 is amenable to virtually all DFO methods, making it unclear which of the algorithms to employ. Methods which have been trialled include: the finite difference based ZOO attack (Chen et al., 2017), a combinatorial direct search of the perturbation ε∞ energy constraint method COMBI (Moon et al., 2019), a genetic direct search method GenAttack (Alzantot et al., 2017), a combinatorial direct search of the perturbation energy constraint method COMBI (Moon et al., 2019), and most recently a randomized direct-search method (Andriushchenko et al., 2019). Notably missing from the aforementioned list are model-based methods.

Given a set of q samples Ψ = {y1, ..., yq} with y ∈ R^n, model-based DFO methods start by identifying the minimiser of the objective among the samples at iteration k, x^k = arg min_y∈Ψ L(y). Following this, a model for the objective function L is constructed, typically centred around the minimizer. In its simplest form one uses a polynomial approximation to the objective, such as a quadratic model centred in x^k

\[ m(x^k + p) = a + c^T p + \frac{1}{2} p^TMp, \]

with a, c, p ∈ R^n, and M ∈ R^{n×n} being also symmetric. In a white-box setting one would set c = ∇L(x^k) and M = ∇^2L(x^k), but this is not feasible in the black-box setting as we do not have access to the derivatives of the objective function. Thus c and M are usually defined by imposing interpolation conditions

\[ m_k(y^i) = L(y^i) \quad \text{for} \quad i = 1, 2, ..., q, \]

and when q < 1 + n + n(n + 1)/2 (i.e. the system of equations is under-determined) other conditions are introduced according to which algorithm is considered. The objective model (3) is considered to be a good estimate of the objective in a neighbourhood referred to as a trust region. Once the model m_k is generated, the update step p is computed by solving the trust region problem

\[ \min_p m_k(x_k + p), \quad \text{subject to} \quad \|p\| \leq \Delta, \]

where Δ is the radius of the region where we believe the model to be accurate, for more details see [Nocedal & Wright, 2006]. The new point x_k + p is added to Ψ and a prior point is potentially removed. Herein we consider an exemplary model-based method called BOBYQA.

**BOBYQA** The BOBYQA algorithm, introduced in (Powell, 2009), updates the parameters of the model a, c, and M, in each iteration in such a way as to minimise the change in the quadratic term M_k between iterates while otherwise fitting the sample values:

\[ \min_{a_k, c_k, M_k} \|M_k - M_{k-1}\|_F^2 \]

\[ \text{s.t.} \quad m_k(y^i) = L(y^i), \quad i = 1, 2, ..., q, \]

with n + 1 < q < 1 + n + n(n + 1)/2 and M_k initialised as the zero matrix. When the number of parameters q = n + 1 then the model is considered as linear with M_k set as zero. We further allow only κ queries at each implementation of BOBYQA, since after the model is generated few iterations are needed to find the minimum.

### 3.1. Computational Scalability and Efficiency

For improved computational scalability and efficiency, we do not solve (1) for η ∈ R^n directly, but instead use domain sub-sampling and hierarchical liftings: domain sub-sampling iteratively sweeps over batches of b < n variables, see [3], while hierarchical liftings clusters and perturb variables simultaneously, see [12].

**Domain Sub-Sampling** The simplest version of domain sub-sampling consists of partitioning input dimension n into smaller disjoint domains; for example, k = [n/b] domains Ω^j of size b ≪ n which are disjoint and which cover all of [n]. Rather than solving (1) for η ∈ R^n directly, for each of j = 1, ..., k one sequentially solves for η^j ∈ R^n which are only non-zero for entries in Ω^j. The resulting sub-domain perturbations η^j are then summed to generate the full perturbation η = Σ^k_j=1 η^j, see Figure 3 as an example. That is, the optimisation problem (1) is adapted to repeatedly looping over j = 1, ..., k:

\[ \min_{\eta^j} \mathcal{L} \left( X + \sum_{h \neq j} \eta^h, \eta^j \right) \quad \text{s.t.} \quad \sum_{h=1}^{k} \|\eta^h\|_{∞} \leq \varepsilon \]

\[ \left[ X + \sum_{h=1}^{k} \eta^h \right]_r \geq l; \quad \left[ X + \sum_{h=1}^{k} \eta^h \right]_r \leq u \quad \forall \eta \in \Omega^j \]

where the Ω^j may be reinitialised; in particular following each loop over R^n which occurs at j = k.

We considered three possible ways of selecting the domains

BOBYQA was selected among the numerous types of model-based DFO algorithms due to its efficiency observed for other similar problems requiring few model samples as in climate modelling (Tett et al., 2013).
and the level perturbations \( \eta \) evolves through the iterations when an image in \( \mathbb{R}^{4 \times 4} \) is attacked. In (a) the perturbation is \( \eta = \eta^0 \) and we select a sub-domain of \( b = 4 \) pixels (in red). Once we have found the optimal perturbation \( \eta^1 \) in the selected sub-domain, we update the perturbation in (b) and select a new sub-domain of dimension \( b \). The same is repeated in (c).

In Figure 3 we compare how these different sub-sampling techniques perform when generating adversarial example for the MNIST and CIFAR10 dataset. It can be observed that variance sampling consistently performs better than random and ordered sampling. This suggest that pixels belonging to high-contrast regions are more influential than the ones in a low-contrast one, and hence variance sampling is the preferable ordering.

**Hierarchical Lifting** When the domain is very high dimensional, working on single pixels is not efficient as the above described method would imply modifying only a very small proportion of the image; for instance, we will choose \( b = 50 \) even when \( n \) is almost three-hundred-thousand. Thus to perturb wider portions of the image, we consider a hierarchy of liftings as in the ZOO attack presented in Chen et al., 2017. We seek an adversarial example by optimising over increasingly higher dimensional spaces at each step referred here as level \( \ell \) lifted to the image space. As an illustration, Figure 4 shows that hierarchical lifting has a significant impact on the minimisation of the loss function.

At each level \( \ell \) we consider a linear lifting \( D^\ell : \mathbb{R}^{m_\ell} \rightarrow \mathbb{R}^n \) and find a level perturbation \( \hat{\eta}_\ell \in \mathbb{R}^{m_\ell} \) which is added to the full perturbation \( \eta \), according to

\[
\eta = \sum_{j=0}^{\ell} \eta_j = \sum_{j=0}^{\ell} D^\ell \hat{\eta}_j,
\]

where \( \eta_0 \) is initialised as 0 and the level perturbations \( \eta_j \) of the previous layers are considered as fixed. Moreover, we impose that at each level, the grid has to double in refinement, i.e. \( m_{\ell+1} = 4m_\ell \). An example of how this works is illustrated in Figure 5.

When generating our adversarial examples, we considered
Two kind of liftings. The first kind of liftings is based on interpolation operations; a sorting matrix $S^L : \mathbb{R}^{m \times \ell} \rightarrow \mathbb{R}^n$ is applied such that every index of $\hat{\eta}$ is uniquely associated to a node of a coarse grid masked over the original image. Afterwards, an interpolation $L^R : \mathbb{R}^n \rightarrow \mathbb{R}^{n}$ is implemented over the values in the coarse grid, i.e. $\eta = L^RS^L \hat{\eta}$.

The second kind of liftings, instead, forces the perturbation to high-frequency since there is several literature on these perturbations being the most effective (Quo et al., 2018; Gopalakrishnan et al., 2018; Sharma et al., 2019).

Some preliminary results lead us to consider the “Block” lifting which considers a piecewise constant interpolation and corresponds to the one also used in (Moon et al., 2019). Alternative piecewise linear or randomised orderings were also tried, but found not to be appreciably better to justify the added complexity. As we show for the example in Figure 6, this interpolation lifting divides an image in disjoint blocks via a coarse grid and associates to each of the blocks the same value of a parameter in $\eta_L$. We characterise the lifting $D^L$ with the following conditions

$$\sum_{j} D^L_{i,j} = 1 \quad \forall i \in \{1, \ldots, n\} \quad (10)$$

$$\sum_{i} D^L_{i,j} = n/m \quad \forall j \in \{1, \ldots, m\}. \quad (11)$$

Since $m\ell$ may still be very high (usually $m_d = n$), for each level $\ell$ we apply domain sub-sampling and consider $\hat{\eta} = \sum_{j=0}^{k} \hat{\eta}^j$. We order the blocks according to the variance of mean intensity among neighbouring blocks, in contrast to the variance within each block which was suggested in (Chen et al., 2017). Consequently, at each level the adversarial example is found by solving the following iterative problem

$$\min_{\hat{\eta}^\ell} \mathcal{L}\left(X + \hat{\eta}, \Omega^k \bar{\eta}^\ell \right) \quad s.t. \quad \left\| \hat{\eta} + \Omega^k \bar{\eta}^\ell \right\|_\infty \leq \varepsilon_\infty \quad (12)$$

$$\left[ X + \hat{\eta} + \Omega^k \bar{\eta}^\ell \right]_r \geq l \quad \forall r \in \{1, \ldots, n\}$$

$$\left[ X + \hat{\eta} + \Omega^k \bar{\eta}^\ell \right]_r \leq u \quad \forall r \in \{1, \ldots, n\},$$

where $\hat{\eta} = \sum_{i=0}^{\ell-1} \eta_i + \Omega^{\ell} \sum_{m \neq \ell} \bar{\eta}^m$.

In its simplest formulation, hierarchical lifting struggles with the pixel-wise interval constraint. $X + \eta \in [l, u]^n$. To address this we allow the entries in $\hat{\eta}$ to exceed the interval and then reproject the pixel-wise entries into the interval.

### 3.2. Algorithm pseudo-code

Our BOBYQA based algorithm is summarised in Algorithm 1, note that not using the hierarchical method corresponds to having one level with $m = n$. A Python implementation of the proposed algorithm based on BOBYQA package from (Cartis et al., 2019) is available on Github.

### 4. Comparison of Derivative Free Methods

We compare the performance of our BOBYQA based algorithm to GenAttack (Alzantot et al., 2019), combinatorial attacks COMBI (Moon et al., 2019) and SQUARE (An-driushchenko et al., 2019). The performance is measured by considering the distribution of queries needed to successfully find adversaries to different networks trained on three standard datasets: MNIST (Leun et al., 1998), CIFAR10 (Krizhevsky, 2009), and ImageNet (Deng et al., 2009).

#### 4.1. Parameter Setup for Algorithms

Our experiments rely for GenAttack (Alzantot et al., 2019), COMBI (Moon et al., 2019), and SQUARE (An...
A Model-Based Derivative-Free Approach to Black-Box Adversarial Examples: BOBYQA

Algorithm 1 BOBYQA Based Algorithm

1: \textbf{Input:} Image $X \in \mathbb{R}^n$, target label $t$, maximum perturbation $\varepsilon_\text{max}$, Neural Net $F$, initial hierarchical level dimensions $m_1$, maximum number of evaluations $n_\text{max}$, batch sampling size $b$, and maximum number $k$ of queries that we are allowed to do for each batch.

2: \textbf{Initialise} $\eta \leftarrow 0 \in \mathbb{R}^n$, $n_\text{eval} = 0$, $\ell = 1$.

3: \textbf{while} $\arg \max F(X + \eta) \neq t$ and $n_\text{eval} < n_\text{max}$ \textbf{do}

4: \hspace{0.5cm} Compute the number of sub samplings necessary to cover the whole domain $n_{\text{sub}} = n/m$

5: \hspace{0.5cm} Generate the lifting matrix $D_\ell$

6: \hspace{0.5cm} for $j = 1, \ldots, n_{\text{sub}}$ do

7: \hspace{1cm} Compute the matrix $\Omega^j$ which selects $b$ dimensions of the $m$-dimensional domain.

8: \hspace{1cm} Define the bounds for a perturbation over the selected pixels of $X + \eta$.

9: \hspace{1cm} Find $\eta^j_\ell$ by implementing the BOBYQA optimisation to the problem [12].

10: \hspace{1cm} Update the noise $\eta^+ = D_\ell \eta^j_\ell$.

11: \hspace{1cm} $n_\text{eval} += \kappa$, $\ell += 1$, $m += 4$.

12: \hspace{0.5cm} end for

13: \hspace{0.5cm} end while

14: \hspace{0.5cm} if $\arg \max F(X + \eta) = t$ then

15: \hspace{1cm} The perturbation is successful.

16: \hspace{0.5cm} else if $n_\text{eval} > n_\text{max}$ then

17: \hspace{1cm} The perturbation was not successful with $n_\text{max}$ iterations.

18: \hspace{0.5cm} end if

\texttt{druishchenko et al., 2019} on publicly available implementations\footnote{GenAttack: https://github.com/nesl/adversarial_genattack. COMBI: https://github.com/snu-mllab/parsimonious-blackbox-attack. SQUARE: https://github.com/max-andr/square-attack.} with same hyperparameter setting and hierarchical approach as suggested by the respective authors.

For the proposed algorithm based on BOBYQA, we tuned three main parameters: the dimension of the initial set $q$, the batch dimension $b$, and the trust region radius.

\textbf{Batch Dimension} Figure [7] shows the loss value averaged over 20 images for attacks to NNs trained on CIFAR10, and ImageNet datasets when different batch dimensions are chosen. The average objective loss as a function of network queries is largely insensitive to the batch sizes, but with modest differences for the larger ImageNet data set where $b = 25$ was observed to require modestly fewer queries. For the remained of the simulations we use $b = 50$ as a good trade-off between faster model generation and good performances.

\textbf{Initial Set Dimension} Once a subdomain of dimension $b$ is chosen, the model [3] is initialised with a set of $q$ samples on which the interpolation conditions $[4]$ are imposed. There are two main choices for the dimension of the set: either $q = b + 1$ or $q = 2b + 1$ in an attack to an image of MNIST with $\varepsilon = 0.3$. We chose for both the methods $b = 50$ and a maximum of 30 function evaluations after the model was initialised, i.e. $\kappa = q + 30$.

\textbf{Trust Region Radius} Once the model for the optimisation is built, the step of the optimisation is bounded by the

\begin{figure}[h]
\centering
\begin{subfigure}{0.4\textwidth}
\centering
\includegraphics[width=\textwidth]{cifar10_loss.png}
\caption{CIFAR10}
\end{subfigure}
\begin{subfigure}{0.4\textwidth}
\centering
\includegraphics[width=\textwidth]{imagenet_loss.png}
\caption{ImageNet}
\end{subfigure}
\caption{Comparison in loss function according to the different batch dimensions $b$ and the different dataset. After the linear model is generated, the optimisation algorithm is always allowed to query the net 5 times if $b = 25$ or $b = 50$, or 10 times if $b = 100$. For ImageNet we are using the hierarchical lifting approach.}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{loss_plot.png}
\caption{Comparison on how the loss $L$ decreases when the initial set dimension is either $q = b + 1$ or $q = 2b + 1$ in an attack to an image of MNIST with $\varepsilon = 0.3$. We chose for both the methods $b = 50$ and a maximum of 30 function evaluations after the model was initialised, i.e. $\kappa = q + 30$.}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{optimization_plot.png}
\caption{The Constraint Optimisation by Linear Approximation (COBYLA), a linear based model DFO algorithm, was introduced before BOBYQA [Powell, 2007]; however, COBYLA considers different constraints on the norm of the variable. Because of this and the possibility to extend the method to quadratic models, we name our algorithm after BOBYQA.}
\end{figure}
For the hierarchical lifting approach we consider an initial sub-domain of dimension $m_1 = 4 \times 4 \times 3$, as this is the biggest grid that we can optimise over with a batch $b = 50$. After considering $m_7 = 256 \times 256 \times 3$, we make use of $m_8 = 299 \times 299 \times 3$ and do not consider further levels.

4.2. Dataset and Neural Network Specifications

Experiments on each dataset are performed with one of the best performing NN architectures as described below.

**MNIST/CIFAR10** MNIST and CIFAR10 are two datasets with images divided between 10 classes and of dimension 28x28x1 and 32x32x3 respectively. On them we apply the net introduced in (Chen et al., 2017) which is structured in succession by: 2 Conv layers with ReLu activation followed by a maxpooling layer. This process is repeated twice and then two dense layers with ReLu activation are applied. Finally a softmax layer generates the output vector. For each dataset, we train the same architecture in two different ways obtaining separate nets. One is obtained by optimising
the accuracy of the net on raw unperturbed images, while
the other is trained with the application of the distillation
defence by [Papernot et al., 2016].

To generate a comprehensive distribution for the queries
at each energy budget, for both the two trained nets and
10 images per class, we attempt to misclassify an image
tARGETING all of the 9 remaining classes; this way we generate
a total of 900 perturbations per energy budget. For these
two datasets the images are of relative low dimension and
we do not apply the hierarchical approach.

ImageNet This is a data-set of millions of images with
a dimension of $299 \times 299 \times 3$ divided between 1000 classes.
For this data-set we consider the Inception-v3 net [Szegedy
et al., 2016] trained with and without the adversarial defence
proposed in [Kurakin et al., 2016]. Due to the large number
of target classes in ImageNet, we perform tests on random
images and target classes. The number of tests conducted
for Inception-v3 [Szegedy et al., 2016] and the adversarially
trained variant [Kurakin et al., 2016] are: 303 and 120 for
$\epsilon_\infty = 0.05$, 155 and 114 for $\epsilon_\infty = 0.02$ and 149 and 116
for $\epsilon_\infty = 0.01$ respectively.

4.3. Experimental Results

In Figure 9 we present the cumulative fraction of images
mislabeled (bridged by CDF for cumulative distribution
function) as a function of the number of queries to the NN
for different perturbation energies $\epsilon_\infty$. The pixels are nor-
malised to be in the interval $(-1/2, 1/2)$, hence, $\epsilon_\infty = 0.1$
would imply that any pixel is allowed to change 10% of the
total intensity range from its initial value. By illustrating the
CDFs we easily see which method has been able to misclas-
sify the largest fraction of images in the given test-set for a
fixed number of queries to the NN. It can be observed that the
proposed BOBYQA based approach achieves state-of-
the-art results when the perturbation bound of $\eta$ decreases.
This behaviour is consistent across all of the considered
datasets (MNIST, CIFAR10, and ImageNet); however, the energy
at which the BOBYQA algorithm performs the best, varies in each case.

In the experiments we also considered nets trained with
defence methods, distillation [Papernot et al., 2016] for
MNIST and CIFAR10 datasets while adversarial training
[Kurakin et al., 2016] for ImageNet, and the results can be
identified in Figure 9 by the dashed lines. Similar to the pre-
vious case, we observe that the proposed BOBYQA based
algorithm performs the best when the energy perturbation
decreases. Moreover, the BOBYQA based algorithm seems
to be the least affected in its performance when the any
defence is used; for example, at $\epsilon_\infty = 0.01$ and 15,000 queries,
the defence reduces the CDF of COMBI by 0.078 compared
to 0.051 for BOBYQA. This further supports the idea that
for more challenging scenarios model-based approaches are
preferable as compared to model-free counterparts.

We associate the counter-intuitive improvement of the CDF
in the MNIST and ImageNet with high perturbation ener-
gies cases to the distillation and the adversarial training
being focused primarily on low energy perturbations. For
ImageNet, non-model-based algorithms use different hier-
archical approaches which we expect leads in part to the
superior performance of COMBI in Fig. 9 panels (g)-(l).

5. Discussion and Conclusion

We have introduced BOBYQA, a method to search adver-
sarial examples based on a model-based DFO algorithm and
have conducted some experiments to understand how it com-
pares to existing GenAttack [Alzantot et al., 2019], COMBI
(Moon et al., 2019), and SQUARE (Andriushchenko et al.,
2019) attack, when targeted black-box adversarial examples
are searched with the fewest queries to a neural net.

Following the results of the experiments that we presented
above, the method with which generating the adversarial
example should be chosen according to which setting the
adversary is considering. When the perturbation energy is
high, one should choose either COMBI if the input is high-
dimensional or SQUARE if the input is low-dimensional.
On the other hand, a model-based approach like BOBYQA
should be considered as soon as the complexity of the setting
increases, e.g. the maximum perturbation energy is reduced
or the net is adversarially trained.

With the BOBYQA attack algorithm we have introduced a
different approach for the generation of targeted adversarial
examples in a black-box setting with the aim of exploring
what advantages are achieved by considering model-based
DFO algorithms. We did not focus on presenting an al-
gorithm which is in absolute the most efficient; primarily
because our algorithm has several aspects in which to be
improved. The BOBYQA attack is limited by the imple-
mentation of py-BOBYQA [Cartis et al., 2019] since the
element-wise constraints do not allow the consideration of
more sophisticated liftings which leverage on compressed
sensing, to name one of the many possible variations.

In conclusion, the results in this paper support how sophisti-
cated misclassification methods are preferable in challeng-
ing settings. As a consequence, variations on our model-
based algorithms should be considered in the future as a tool
to establish the effectiveness of newly presented adversarial
defence techniques.

Acknowledgements

This publication is based on work supported by the EPSRC Centre for Doctoral Training in Industrially Focused Mathematical Modelling (EP/L015803/1) in collaboration with New Rock Capital Management.

References

Alzantot, M., Sharma, Y., Chakraborty, S., Zhang, H., Hsieh, C.-J., and Srivastava, M. B. Genattack: Practical black-box attacks with gradient-free optimization. In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO), pp. 1111–1119, July 2019. doi: 10.1145/3321707.3321749.

Andriushchenko, M., Croce, F., Flammarion, N., and Hein, M. Square attack: a query-efficient black-box adversarial attack via random search. arXiv preprint arXiv:1912.00049, 2019.

Carlini, N. and Wagner, D. Towards evaluating the robustness of neural networks. In Proceedings of the IEEE Symposium on Security and Privacy (SP), pp. 39–57, May 2017. doi: 10.1109/SP.2017.49.

Cartis, C., Fiala, J., Marteau, B., and Roberts, L. Improving the flexibility and robustness of model-based derivative-free optimization solvers. ACM Trans. Math. Softw., 45 (3), August 2019. doi: 10.1145/3338517.

Chen, P.-Y., Zhang, H., Sharma, Y., Yi, J., and Hsieh, C.-J. ZOO: zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In Proceedings of the ACM Workshop on Artificial Intelligence and Security (AISec), pp. 15–26, November 2017. doi: 10.1145/3128572.3140448.

Chen, P.-Y., Sharma, Y., Zhang, H., Yi, J., and Hsieh, C.-J. Ead: elastic-net attacks to deep neural networks via adversarial examples. In Proceedings of the AAAI Conference on Artificial Intelligence, pp. 10–17, February 2018.

Conn, A. R., Scheinberg, K., and Vicente, L. N. Introduction to derivative-free optimization, volume 8. SIAM, 2009.

Dalvi, N., Domingos, P., Sanghavi, S., Verma, D., et al. Adversarial classification. In Proceedings of the ACM International conference on Knowledge Discovery and Data Mining (SIGKDD), pp. 99–108, August 2004. doi: 10.1145/1014052.1014066.

Deng, J., Dong, W., Socher, R., Li, L., Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 248–255, June 2009. doi: 10.1109/CVPR.2009.5206848.

Dhillon, G. S., Azizzadenesheli, K., Lipton, Z. C., Bernstein, J., Kossaifi, J., Khanna, A., and Anandkumar, A. Stochastic activation pruning for robust adversarial defense. In Proceedings of the International Conference on Learning Representations (ICLR), April 2018.

Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., Prakash, A., Kohno, T., and Song, D. Robust physical-world attacks on deep learning visual classification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1625–1634, June 2018. doi: 10.1109/CVPR.2018.00175.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. In Proceedings of the International Conference on Learning Representations (ICLR), May 2015.

Gopalakrishnan, S., Marzi, Z., Madhow, U., and Pedarsani, R. Toward robust neural networks via sparsification. arXiv preprint arXiv:1810.10625, 2018.

Guo, C., Frank, J. S., and Weinberger, K. Q. Low frequency adversarial perturbation. In Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI), July 2018.

He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 1026–1034, December 2015. ISBN 9781467383912. doi: 10.1109/ICCV.2015.123.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., and Kingsbury, B. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal Processing Magazine, 29(6):82–97, November 2012. doi: 10.1109/MSP.2012.2205597.

Ilyas, A., Engstrom, L., Athalye, A., and Lin, J. Black-box adversarial attacks with limited queries and information. In Proceedings of the International Conference on Machine Learning (ICML), pp. 2137–2146, July 2018.

Krizhevsky, A. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Kurakin, A., Goodfellow, I., and Bengio, S. Adversarial machine learning at scale. arXiv preprint arXiv:1611.01236, 2016.
Kurakin, A., Goodfellow, I., and Bengio, S. Adversarial examples in the physical world. In *Proceedings of the International Conference on Learning Representations (ICLR), Workshop Track*, April 2017.

Larson, J., Menickelly, M., and Wild, S. M. Derivative-free optimization methods. *Acta Numerica*, 28:287–404, 2019. doi: 10.1017/S0962492919000060.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998. doi: 10.1109/5.726791.

Monti, F., Frasca, F., Eynard, D., Mannion, D., and Bronstein, M. M. Fake news detection on social media using geometric deep learning. *arXiv preprint arXiv:1902.06673*, 2019.

Moon, S., An, G., and Song, H. O. Parsimonious black-box adversarial attacks via efficient combinatorial optimization. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 4636–4645, June 2019.

Narodytska, N. and Kasiviswanathan, S. Simple black-box adversarial attacks on deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1310–1318, July 2017. doi: 10.1109/CVPRW.2017.172.

Nocedal, J. and Wright, S. J. *Numerical Optimization*. Springer-Verlag New York, 2006. ISBN 978-0-387-30303-1. doi: 10.1007/978-0-387-40065-5.

Papernot, N., McDaniel, P., Wu, X., Jha, S., and Swami, A. Distillation as a defense to adversarial perturbations against deep neural networks. In *Proceedings of the IEEE Symposium on Security and Privacy (SP)*, pp. 582–597, May 2016. doi: 10.1109/SP.2016.41.

Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., and Swami, A. Practical black-box attacks against machine learning. In *Proceedings of the ACM on Asia Conference on Computer and Communications Security (ASIA CCS)*, pp. 506–519, April 2017. doi: 10.1145/3052973.3053009.

Powell, M. J. A view of algorithms for optimization without derivatives. *Mathematics Today-Bulletin of the Institute of Mathematics and its Applications*, 43(5):170–174, 2007.

Powell, M. J. The bobyqa algorithm for bound constrained optimization without derivatives. Technical Report DAMTP 2009/NA06, University of Cambridge, 2009.

Sharma, Y., Ding, G. W., and Brubaker, M. On the effectiveness of low frequency perturbations. *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 3389–3396, August 2019. doi: 10.24963/ijcai.2019/470.

Sitawarin, C., Bhagoji, A. N., Mosenia, A., Chiang, M., and Mittal, P. Darts: Deceiving autonomous cars with toxic signs. *arXiv preprint arXiv:1802.06430*, 2018.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, April 2014.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, June 2016. doi: 10.1109/CVPR.2016.308.

Tett, S. F. B., Mineter, M. J., Cartis, C., Rowlands, D. J., and Liu, P. Can top-of-atmosphere radiation measurements constrain climate predictions? part i: Tuning. *Journal of Climate*, 26(23):9348–9366, November 2013. doi: 10.1175/JCLI-D-12-00595.1.

Tu, C.-C., Ting, P., Chen, P.-Y., Liu, S., Zhang, H., Yi, J., Hsieh, C.-J., and Cheng, S.-M. Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence: Special Technical Track: AI for Social Impact*, July 2019. doi: 10.1609/aaai.v33i01.3301742.

Wang, X., Wang, S., Chen, P.-Y., Wang, Y., Kulis, B., Lin, X., and Chin, S. Protecting neural networks with hierarchical random switching: Towards better robustness-accuracy trade-off for stochastic defenses. In *Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI*, August 2019.

Yuan, X., He, P., Zhu, Q., and Li, X. Adversarial examples: Attacks and defenses for deep learning. *IEEE Transactions on Neural Networks and Learning Systems*, 30(9):2805–2824, September 2019. doi: 10.1109/TNNLS.2018.2886017.