Abstract

We focus on the problem of black-box adversarial attacks, where the aim is to generate adversarial examples for deep learning models solely based on information limited to output labels (hard label) to a queried data input. We use Bayesian optimization (BO) to specifically cater to scenarios involving low query budgets to develop efficient adversarial attacks. Issues with BO’s performance in high dimensions are avoided by searching for adversarial examples among low-frequency vectors from the Fast Fourier Transform (FFT) basis. Our proposed approach achieves better performance to state of the art black-box adversarial attacks that require orders of magnitude more queries than ours.

1. Introduction

Neural networks are now well-known to be vulnerable to adversarial examples: additive perturbations that, when applied to the input, change the network’s output classification (Goodfellow et al., 2014). Work investigating this lack of robustness to adversarial examples often takes the form of a back-and-forth between newly proposed adversarial attacks, methods for quickly and efficiently crafting adversarial examples, and corresponding defenses that modify the classifier at either training or test time to improve robustness. The most successful adversarial attacks use gradient-based optimization methods (Goodfellow et al., 2014; Madry et al., 2017), which require complete knowledge of the architecture and parameters of the target network; this assumption is referred to as the white-box attack setting. Conversely, the more realistic black-box setting requires an attacker to find an adversarial perturbation without such knowledge:

information about the network can be obtained only through querying the target network, i.e., supplying an input to the network and receiving the corresponding output. In terms of information obtained from queries, it can further categorized into soft-label and hard-label. As far as soft-label information is concerned, the information for a query is typically in terms of the logit values or the evaluation of the loss function at that particular input. The more realistic and challenging of the two, i.e., hard-label information obtained from a query is just the label of the input fed into the network.

In real-world scenarios, it is extremely improbable for an attacker to have unlimited bandwidth to query a target classifier. In evaluation of black box attacks, this constraint is usually formalized via the introduction of a query budget: a maximum number of queries allowed to query the model per input, after which an attack is considered to be unsuccessful. Several recent papers have proposed attacks specifically to operate in this query-limited context (Ilyas et al., 2019; 2018; Chen et al., 2017; Tu et al., 2019; Moon et al., 2019; Cheng et al., 2018; 2020); nevertheless, these papers typically consider query budgets in the order of 10,000 or 100,000. This leaves open questions as to whether black-box attacks can successfully attack a deep learning based classifier in severely query limited settings, e.g., with query budgets below 1000. Furthermore, restricting the information available from querying to be hard-label only, makes the aforementioned direction even more challenging. In such a query limited regime, it is natural for an attacker to use the entire query budget, so we ask the pertinent question: In a constrained query limited setting, can one design query efficient yet successful black box adversarial attacks, where the queried information is restricted to being hard-label?

This work proposes a hard-label black-box attack method grounded in Bayesian optimization (Jones et al., 1998; Frazier, 2018), which has emerged as a state of the art black-box optimization technique in settings where minimizing the number of queries is of paramount importance. Straightforward application of Bayesian optimization to the problem of finding adversarial examples is not feasible: the input dimension of even a small neural network-based image classifier is orders of magnitude larger than the standard use

1College of Information and Computer Sciences, University of Massachusetts Amherst, Amherst MA, 01002, USA
2Bosch Center for AI, Pittsburgh PA, 15222, USA
3Computer Science Department, Carnegie Mellon University, Pittsburgh PA, 15213, USA. Correspondence to: Satya Narayan Shukla <seshukla@cs.umass.edu>.

Preliminary work.
Within the black-box setting, adversarial attacks can be performed both via a random gradient-free method, and Sign-OPT attack (Cheng et al., 2018), which refines this approach by replacing binary searches with optimization via sign SGD. In Boundary Attack (Brendel et al., 2017), attacks are generated via random walks along the decision boundary with rejection sampling. Several other attacks have extended this work and refined its query efficiency: HopSkipJumpAttack (Chen et al., 2019) does so with improved gradient estimation, while Guessing Smart (Brunner et al., 2019) incorporates low-frequency perturbations, region masking, and gradients from a surrogate model. In both cases, significant issues remain: the former still requires queries numbering above 10,000 to produce adversarial examples with small perturbations, and the latter relies on resources required to train a surrogate model.

Most work in black-box adversarial attacks has been dedicated to score-based or soft label attacks, where queries return the entire output layer of the network, either as logits or probabilities. Relative to hard-label attacks, queries in the soft label setting receive a large amount of information from each query, making them amenable to approaches from a wide variety of optimization fields and techniques. The most popular avenue is maximizing the network loss with zeroth-order methods via derivative-free gradient estimation, such as those proposed in Ilyas et al. (2019), which uses time-dependent and data-dependent priors to improve the estimate, as well as Ilyas et al. (2018), which estimates gradients using natural evolution strategies (NES). Other methods search for the best perturbation outside of this paradigm; Moon et al. (2019) casts the problem of finding an adversarial perturbation as a discrete optimization problem and uses local search methods to solve. These works all search for adversarial perturbations within a search space with a hard constraint on perturbation size; others (Chen et al., 2017; Tu et al., 2019; Guo et al., 2019) incorporate a soft version of this constraint and perform coordinate descent or random walks to decrease the perturbation size while ensuring the perturbed image is misclassified.

A separate class of transfer-based attacks train a second, fully-observable substitute network, attack this network with white-box methods, and transfer these attacks to the original target network. These may fall into one of the preceding categories or exist outside of the distinction: in Papernot et al. (2016), the substitute model is built with score-based queries to the target network, whereas Liu et al. (2016) trains an ensemble of models without directly querying the target network at all. These methods come with their own drawbacks: they require training a substitute model, which may be costly or time-consuming; attack success is frequently dependent on similarity between substitute and target networks; and overall attack success tends to be lower than that of gradient-based methods.

Beyond these categories, we note that our method here sits among several recent works that find improved success rates and query efficiency from restricting their search for adversarial perturbations to particular low-dimensional subspaces. One common approach is to generate adversarial perturbations from low-frequency noise, such as in Guo et al. (2018), which improves existing attacks (Ilyas et al., 2018; Brendel et al., 2017) by confining searches to subspaces of low

---

**Hard Label Black-box Adversarial Attacks in Low Query Budget Regimes**

---

**2. Related Work**

Within the black-box setting, adversarial attacks can be further categorized by the exact nature of the information received from a query. This work exists in the restrictive hard-label or decision-based setting, where queries to the network yield only the predicted class of the input, with no other information about the network’s final layer output. The most successful work in this area is OPT attack (Cheng et al., 2018), which reformulates the problem as a search for the direction of the nearest decision boundary and employs a random gradient-free method, and Sign-OPT attack (Cheng et al., 2020), which refines this approach by replacing binary searches with optimization via sign SGD. In Boundary Attack (Brendel et al., 2017), attacks are generated via random walks along the decision boundary with rejection sampling. Several other attacks have extended this work and refined its query efficiency: HopSkipJumpAttack (Chen et al., 2019)
frequency basis vectors in the Discrete Cosine Transform (DCT) basis, and in Brunner et al. (2019), which employs a Perlin noise basis. In a similar vein, Ilyas et al. (2019) exhibits that local spatial similarity of images, i.e., the tendency of nearby pixels to be similar, extends to gradients, and uses this observation to motivate focusing on tile-based perturbations.

Finally, there has been some recent interest in leveraging Bayesian optimization (BO) for constructing adversarial perturbations. For example, Zhao et al. (2019) uses BO to solve the δ-step of an alternating direction of method multipliers (ADMM) approach, Co et al. (2018) searches within a set of procedural noise perturbations using BO and Gopakumar et al. (2018) uses BO to find maximal distortion error by optimizing perturbations defined using 3 parameters. On the other hand, prior work in which Bayesian optimization plays a central role, the use cases and experiments are performed only in relatively low-dimensional settings, highlighting the main challenge of its application: Suya et al. (2017) examines an attack on a spam email classifier with 57 input features, and in Co (2017) image classifiers are attacked but notably the attack does not scale beyond MNIST classifiers. In contrast to these past works, the main contribution of this paper is to show that Bayesian Optimization presents as a scalable, query-efficient alternative for large-scale hard-label black-box adversarial attacks when combined with searching in structured low dimensional subspaces and employing mapping techniques to get the final adversarial perturbation.

3. Problem Formulation

The following notation and definitions will be used throughout the remainder of the paper. Let $F$ be the target neural network. We assume that $F : \mathbb{R}^d \rightarrow \{1, \ldots, K\}$ is a $K$-class image classifier that takes normalized RGB inputs: $x \in \mathbb{R}^d$ where each channel of each pixel is bounded between 0 and 1, $y \in \{1, \ldots, K\}$ denotes the original label, and the corresponding output $F(x)$ is the final output label or class.

Rigorous evaluation of an adversarial attack requires careful definition of a threat model: a set of formal assumptions about the goals, knowledge, and capabilities of an attacker (Carlini & Wagner, 2017). We assume that, given a correctly classified input image $x$, the goal of the attacker is to find a perturbation $\delta$ such that $x + \delta$ is misclassified, i.e., $F(x + \delta) \neq F(x)$. We operate in the hard-label black-box setting, where we have no knowledge of the internal workings of the network, and a query to the network $F$ yields only the final output class (top-1 prediction). To enforce the notion that the adversarial perturbation should be small, we take the common approach of requiring that $\|\delta\|_p$ be smaller than a given threshold $\epsilon$ in some $\ell_p$ norm. In this work, we specifically focus on $\ell_2$ and $\ell_\infty$ norm. Finally, we let $t$ denote the query budget, i.e., if an adversarial example is not found after $t$ queries to the target network, the attack is considered to be unsuccessful.

In line with most adversarial attack setups, we pose the attack as a constrained optimization problem, defined below:

$$\max_{\delta} f(x, y, \delta)$$

subject to $\|\delta\|_p \leq \epsilon$ and $(x + \delta) \in [0, 1]^d$, 

where $f(x, y, \delta) = \begin{cases} 0 & \text{if } F(x + \delta) \neq y \\ -1 & \text{if } F(x + \delta) = y \end{cases}$

Crucially, the input $x + \delta$ to $f$ is an adversarial example for $F$ if and only if $f(x, y, \delta) = 0$.

Though our objective function is straightforward, we empirically show that it leads to significant performance improvements over the current state of the art for hard-label black-box attacks for both $\ell_\infty$ and $\ell_2$ threat models.

We briefly note that the above threat model and objective function were chosen for simplicity and for ease of directly comparing with other black box attacks, but the attack method we propose is compatible with many other threat models. For example, we may change the goals of the attacker or measure $\delta$ in $\ell_1$ norm instead of $\ell_2$ and $\ell_\infty$ norms with appropriate modifications to the objective function and constraints in equation 1.

4. Proposed Attack Method

In this section, we present the proposed method for solving the optimization problem in equation 1. We begin with a brief description of Bayesian optimization (Jones et al., 1998) followed by its application to generating black-box adversarial examples. Finally, we describe our method for attacking a classifier with high input dimension (e.g. ImageNet) in a query-efficient manner.

4.1. Bayesian Optimization

Bayesian Optimization (BO) is a method for black box optimization particularly suited to problems with low dimension and expensive queries. Bayesian Optimization consists of two main components: a Bayesian statistical model and an acquisition function. The Bayesian statistical model, also referred to as the surrogate model, is used for approximating the objective function: it provides a Bayesian posterior probability distribution that describes potential values for the objective function at any candidate point. This posterior distribution is updated each time we query the objective function at a new point. The most common surrogate model
for Bayesian optimization are Gaussian processes (GPs) (Rasmussen & Williams, 2005), which define a prior over functions that are cheap to evaluate and are updated as and when new information from queries becomes available. We model the objective function $h$ using a GP with prior distribution $\mathcal{N}(\mu_0, \Sigma_0)$ with constant mean function $\mu_0$ and Matern kernel (Shahriari et al., 2016; Snoek et al., 2012) as the covariance function $\Sigma_0$, which is defined as:

$$
\Sigma_0(x, x') = \theta_0^2 \exp(-\sqrt{5}r) \left( 1 + \sqrt{5}r + \frac{5}{3}r^2 \right),
$$

$$
r^2 = \sum_{i=1}^{d'} \frac{(x_i - x'_i)^2}{\theta_i^2},
$$

where $d'$ is the dimension of input and $\{\theta_i\}_{i=0}^{d'}$ and $\mu_0$ are hyperparameters. We select hyperparameters that maximize the posterior of the observations under a prior (Shahriari et al., 2016; Frazier, 2018).

The second component, the acquisition function $A$, assigns a value to each point that represents the utility of querying the model at this point given the surrogate model. We sample the objective function $h$ at $x_n = \arg\max_x A(x|D_{1:n-1})$ where $D_{1:n-1}$ comprises of $n-1$ samples drawn from $h$ so far. Although this itself may be a hard (non-convex) optimization problem to solve, in practice we use a standard approach and approximately optimize this objective using the LBFGS algorithm. There are several popular choices of acquisition function; we use expected improvement (EI) (Jones et al., 1998), which is defined as

$$
\text{EI}_n(x) = \mathbb{E}_n \left[ \max \ (h(x) - h^*_n, 0) \right],
$$

where $\mathbb{E}_n[\cdot] = \mathbb{E}[\cdot | D_{1:n-1}]$ denotes the expectation taken over the posterior distribution given evaluations of $h$ at $x_1, \cdots, x_{n-1}$, and $h^*_n$ is the best value observed so far.

Bayesian optimization framework as shown in Algorithm 2 runs these two steps iteratively for the given budget of function evaluations. It updates the posterior probability distribution on the objective function using all the available data. Then, it finds the next sampling point by optimizing the acquisition function over the current posterior distribution of GP. The objective function $h$ is evaluated at this chosen point and the whole process repeats.

In theory, we may apply Bayesian optimization directly to the optimization problem in equation 1 to obtain an adversarial example, stopping once we find a point where the objective function reaches 0. In practice, Bayesian optimization’s speed and overall performance fall dramatically as the input dimension of the problem increases. This makes running Bayesian optimization over high dimensional inputs such as ImageNet (input dimension $3 \times 299 \times 299 = 268203$) practically infeasible; we therefore require a method for reducing the dimension of this optimization problem.

4.2. Bayes-Attack: Generating Adversarial Examples using Bayesian Optimization

Black-box attack methods tend to require a lot of queries because of the search space being high dimensional. The query complexity of these attack methods depends on the adversarial subspace dimension compared to the original image space. These methods can be improved by finding a structured low dimensional subspace so as to make the black-box optimization problem feasible and thereby resulting in adversarial examples with fewer queries and high success rates. In this section, we define two low dimension subspaces favorable for generating $\ell_2$ and $\ell_\infty$ norm constrained hard label black-box attacks.

4.2.1. Low Dimensional Subspace for $\ell_2$ Norm Constrained Attack

To generate $\ell_2$ norm constrained adversarial examples, our attack method utilizes low frequency fast Fourier transform (FFT) basis vectors. FFT is a linear transform which, when applied to a natural image, results in a representation in frequency space by sine and cosine basis vectors. For a given image $x \in \mathbb{R}^{d \times d}$, the output of the FFT transform $X := \text{FFT}(x)$ is defined by

$$
X[u, v] = \frac{1}{d} \sum_{i=0}^{d-1} \sum_{j=0}^{d-1} x[i, j] \exp \left( -j \frac{2\pi}{d} (u \cdot i + v \cdot j) \right)
$$

(2)

where $\frac{1}{d}$ is the normalization constant to obtain isometric transformation, i.e., $\|x\|_2 = \|\text{FFT}(x)\|_2$. The inverse fast fourier transform $x = \text{IFFT}(X)$ is defined by:

$$
x[i, j] = \frac{1}{d} \sum_{u=0}^{d-1} \sum_{v=0}^{d-1} X[u, v] \exp \left( j \frac{2\pi}{d} (u \cdot i + v \cdot j) \right)
$$

(3)

The isometric property holds in reverse direction too, i.e. $\|X\|_2 = \|\text{IFFT}(X)\|_2$. For multi-channel (colored) images, both FFT and IFFT can be applied channel wise independently. The low frequency cosine and sine basis vectors are represented by small $u, v$ values in real and complex components of $X(u, v)$ respectively. To restrict to low-frequencies, we allow only elements in the top-left $[rd] \times [rd]$ square of $X$ to have nonzero entries, where $r \in (0, 1]$ is a parameter that controls how large we allow our search space to be; that is, we enforce $X(u, v) = 0$ if $u > [rd]$ or $v > [rd]$. The adversarial perturbation is then obtained by computing $\text{IFFT}(X)$.

To further reduce the dimension of this search space, we may also omit all sine or all cosine FFT basis vectors by respectively constraining the real or imaginary parts of $X$ to be zero. An ablation study exploring the effect of removing sine or cosine FFT basis vectors on performance is shown in Section 5.4.1.
We describe the complete algorithm in Algorithm 2 where we define the objective function for running the Bayesian optimization in low dimension in Algorithm 1. We let \( \Pi_{B}^r(x_0, \epsilon) \) be the projection on the \( \ell^r \) ball of radius \( \epsilon \) centered at origin. Our method maps the learned perturbation in low dimension subspace back to original input space to obtain the adversarial perturbation. We maintain the \( \ell^p \) norm constraint by projecting the low dimension subspace perturbation on a \( \ell^p \) ball of radius \( \epsilon \). Since, our mapping techniques (IFFT for \( \ell^2 \) and NNI for \( \ell^\infty \)) do not change their respective norms, the final adversarial perturbation obtained after mapping to original input space also follows the \( \ell^p \) constraint.

We describe the complete algorithm in Algorithm 2 where \( x_0 \in \mathbb{R}^{d \times d} \) and \( y_0 \in \{1, \ldots, K\} \) denote the original input image and label respectively. The goal is to learn an adversarial perturbation \( \delta \in \mathbb{R}^{[rd] \times [rd]} \) in much lower dimension, i.e., \( r \ll 1 \). We begin with a small dataset \( D = \{(\delta_1, v_1), \ldots, (\delta_n, v_n)\} \) where each \( \delta_i \) is a \( [rd] \times [rd] \) perturbation sampled from a given distribution and \( v_n \) is the function evaluation at \( \delta_i \), i.e. \( v_n = \text{Obj-Func}(x_0, y_0, \delta_n) \). We iteratively update the posterior distribution of the GP using all available data and query new perturbations obtained by maximizing the acquisition function over the current posterior distribution of GP until we find an adversarial perturbation or run out of query budget. The Bayesian optimization iterations run in low dimension subspace \( [rd] \times [rd] \) but for querying the model we project and map to the original image space and then add the perturbation to the original image as shown in Algorithm 1 to get the perturbed image to conform to the input space of the model. To generate a successful adversarial perturbation, it is necessary and sufficient to have \( v_t \geq 0 \), as described in Section 3. We call our attack successful with \( t \) queries to the model if the Bayesian optimization loop exits after \( t \) iterations (line 12 in Algorithm 2), otherwise it is unsuccessful. Finally, we note that the final adversarial image can be obtained by mapping the learned perturbation back to the original image space and adding to the original image as shown in Figure 1. For multi-channel images, the low dimension subspace is of form \( 3 \times [rd] \times [rd] \) and the whole algorithm works the same way.

In this work, we focus on \( \ell^\infty \) and \( \ell^2 \)-norm perturbations, where the respective projections are defined as:

\[
\Pi_{B}^\infty(x_0, \epsilon) = \min \{ \max\{x_0 - \epsilon, x\}, x_0 + \epsilon\}, \\
\Pi_{B}^2(x_0, \epsilon) = \arg\min_{\|y - x\|_2} \{\|y - x\|_2 \leq \epsilon\}
\]

where \( \epsilon \) is the given perturbation bound.

The initial choice of the dataset \( D \) to form a prior can be done using standard normal distribution, uniform distribution or even in a deterministic manner (e.g. with Sobol sequences).

### 4.2.2. Low Resolution Subspace for \( \ell^\infty \) Norm Constrained Attack

Our method uses a data dependent prior (Ilyas et al., 2019) to reduce the search dimension of the perturbation for generating \( \ell^\infty \) norm constrained adversarial examples. We empirically show that we can utilize the property of spatial local similarity in images to generate adversarial perturbations. For a given image \( x \in \mathbb{R}^{d \times d} \), we search for perturbations in a lower resolution image space \( X \in \mathbb{R}^{[rd] \times [rd]} \) where \( r \in (0, 1] \) and use nearest neighbor interpolation (NNI) \( x' = \text{NNI}(X) \) to obtain the final adversarial perturbation. We note that \( x' \neq x \). The NNI transformation leads to equivalent \( \ell^\infty \) norms, i.e., \( \|X\|_\infty = \|\text{NNI}(X)\|_\infty \). For multi-channel images, NNI can be applied channel-wise independently.

### 4.2.3. Searching in Low Dimension Subspace

We use Bayesian optimization to search for the perturbation in low dimension subspace \( [rd] \times [rd] \) where \( r \in (0, 1] \) and then use the relevant mapping (IFFT for \( \ell^2 \) or and NNI for \( \ell^\infty \)) to obtain the adversarial perturbation in the original image space. This helps in reducing the query complexity for our attack framework, due to the reduction of the search space to a structured low dimensional subspace.

We define the objective function for running the Bayesian optimization in low dimension in Algorithm 1. We let \( \Pi_{B}^{\ell (x_0, \epsilon)} \) be the projection onto the \( \ell^p \) ball of radius \( \epsilon \) centered at origin. Our method maps the learned perturbation in low dimension subspace back to original input space to obtain the adversarial perturbation. We maintain the \( \ell^p \) norm constraint by projecting the low dimension subspace perturbation on a \( \ell^p \) ball of radius \( \epsilon \). Since, our mapping techniques (IFFT for \( \ell^2 \) and NNI for \( \ell^\infty \)) do not change their respective norms, the final adversarial perturbation obtained after mapping to original input space also follows the \( \ell^p \) constraint.

We describe the complete algorithm in Algorithm 2 where \( x_0 \in \mathbb{R}^{d \times d} \) and \( y_0 \in \{1, \ldots, K\} \) denote the original input image and label respectively. The goal is to learn an adversarial perturbation \( \delta \in \mathbb{R}^{[rd] \times [rd]} \) in much lower dimension, i.e., \( r \ll 1 \). We begin with a small dataset \( D = \{(\delta_1, v_1), \ldots, (\delta_n, v_n)\} \) where each \( \delta_i \) is a \( [rd] \times [rd] \) perturbation sampled from a given distribution and \( v_n \) is the function evaluation at \( \delta_i \), i.e. \( v_n = \text{Obj-Func}(x_0, y_0, \delta_n) \). We iteratively update the posterior distribution of the GP using all available data and query new perturbations obtained by maximizing the acquisition function over the current posterior distribution of GP until we find an adversarial perturbation or run out of query budget. The Bayesian optimization iterations run in low dimension subspace \( [rd] \times [rd] \) but for querying the model we project and map to the original image space and then add the perturbation to the original image as shown in Algorithm 1 to get the perturbed image to conform to the input space of the model. To generate a successful adversarial perturbation, it is necessary and sufficient to have \( v_t \geq 0 \), as described in Section 3. We call our attack successful with \( t \) queries to the model if the Bayesian optimization loop exits after \( t \) iterations (line 12 in Algorithm 2), otherwise it is unsuccessful. Finally, we note that the final adversarial image can be obtained by mapping the learned perturbation back to the original image space and adding to the original image as shown in Figure 1. For multi-channel images, the low dimension subspace is of form \( 3 \times [rd] \times [rd] \) and the whole algorithm works the same way.

In this work, we focus on \( \ell^\infty \) and \( \ell^2 \)-norm perturbations, where the respective projections are defined as:

\[
\Pi_{B}^{\infty}(x_0, \epsilon) = \min \{ \max\{x_0 - \epsilon, x\}, x_0 + \epsilon\}, \\
\Pi_{B}^{2}(x_0, \epsilon) = \arg\min_{\|y - x\|_2} \{\|y - x\|_2 \leq \epsilon\}
\]

where \( \epsilon \) is the given perturbation bound.

The initial choice of the dataset \( D \) to form a prior can be done using standard normal distribution, uniform distribution or even in a deterministic manner (e.g. with Sobol sequences).

### 5. Experiments

Our experiments focus on the untargeted attack setting where, given an image correctly classified by the model, the goal is to produce an \( \ell^\infty \)-constrained perturbation such that applying this perturbation to the original image causes misclassification. We evaluate both \( \ell^\infty \) and \( \ell^2 \) norm constrained hard label black-box attacks. We primarily consider performance of Bayes-Attack on ImageNet classifiers and compare its performance to other black-box attacks with respect to success rate over a given query budget. To compare performance, we randomly selected a subset of 1000 images, normalized to \([0, 1]\), from the ImageNet validation set. In all experiments in this section, attacks are performed on this fixed set of images. We primarily compare performance
We treat the size of low dimensional subspace used for running the Bayesian optimization loop as a hyperparameter. For both $\ell_2$ and $\ell_\infty$ attacks on ImageNet, we tune the low dimensional subspace size $|rd| \times |rd|$ over the range $rd \in [5, 18]$. For $\ell_2$ attacks, we treat the option of representing the subspace with sine and cosine FFT basis vectors separately or together as a hyper-parameter. We initialize the GP with $n_0 = 5$ samples drawn from a standard normal distribution in case of $\ell_\infty$ attacks, and from the uniform distribution $[-1, 1]$ for $\ell_2$ attacks. For all the experiments in this section, we use expected improvement as the acquisition function. We also examined other acquisition functions (posterior mean, probability of improvement, upper confidence bound) and observed that our method works equally well with other acquisition functions. We independently tune the hyperparameters on a small validation set and exclude it from our final set of images to be attacked. We used BoTorch\(^2\) packages for implementation.

### 5.2. Untargeted $\ell_2$ attack

We compare the performance of the proposed method Bayes-Attack against OPT attack (Cheng et al., 2018) and Sign-OPT (Cheng et al., 2020), which is the current state of the art among hard-label black-box attacks within the $\ell_2$ threat

---

\(^2\)https://botorch.org/
model. On ImageNet, we attack the pretrained\(^3\) ResNet50 (He et al., 2015), Inception-v3 (Szegedy et al., 2015) and VGG16-bn (Simonyan & Zisserman, 2014).

We compare the performance across three different \(\ell_2\) perturbation bounds, where we set \(\epsilon\) to 5.0, 10.0 and 20.0 respectively. We evaluate the performance of all the methods with a budget of up to 1000 queries. For both OPT attack and Sign-OPT attack, we use the implementations made available by the authors\(^4\) and use hyperparameters as suggested in their respective papers.

Figure 2 considers the the specific case of \(\epsilon = 20.0\) in the above experiment, and exhibits the relationship between success rates and number of queries used for each method. We can see that Bayes-Attack consistently outperforms the other baseline methods for across all query budgets up to 1000. Table 1 compares success rate and average query count of all methods, models, and \(\epsilon\) thresholds for a query budget of 1000. Bayes-Attack not only manages to get attack success rates up to 6\times better than the closest baseline, but simultaneously reduces the average query count by up to a factor of ten.

### 5.3. Untargeted \(\ell_\infty\) attack

We also compare the performance of the proposed method Bayes-Attack against Sign-OPT (Cheng et al., 2020), which is the current state of the art among hard-label black-box attacks within the \(\ell_\infty\) threat model. We use the same query budgets, models, and reported metrics as in the previous subsection. Here, we set the \(\ell_\infty\) perturbation bound \(\epsilon\) to 0.05. Table 2 compares the performance of \(\ell_\infty\) norm constrained attacks in terms of success rate and average query count. The proposed method Bayes-Attack consistently achieves significant performance improvements over Sign-OPT.

### 5.4. Ablation Study

In this section, we perform ablation studies on ImageNet by exploring different low dimensional structured subspaces

---

\(^3\)Pretrained models available at https://pytorch.org/docs/stable/torchvision/models

\(^4\)https://drive.google.com/file/d/10bxFajAHm9FJfLy41TJf1gq45FSiP8E/edit

\(^5\)https://github.com/LeMinhThong/blackbox-attack
and comparing mapping techniques for both $\ell_2$ and $\ell_\infty$ threat models.

5.4.1. LOW DIMENSION SUBSPACES

In case of the $\ell_2$ threat model, we utilize the low dimensional subspace generated by the low frequency sine and cosine FFT basis vectors. We can also separately consider the cosine FFT basis or sine FFT basis separately by using only the real components or imaginary components of the frequency domain representation.

We compare the attacks generated in the low dimension subspace created using cosine and sine FFT basis vectors separately and together. For this experiment, we perform hard label black-box attacks on ResNet50 trained on ImageNet with $\ell_2$ perturbation set to 20.0. We maintain a query budget of 1000 across the experiments. For fair comparison, we keep the size of low dimension subspace same across the experiments, i.e. $3 \times 18 \times 18$.

Table 4 shows the performance of both the mapping techniques on $\ell_\infty$ and $\ell_2$ threat models. FFT basis vectors perform better than nearest neighbor interpolation in the $\ell_2$ threat model, while nearest neighbor performs better for $\ell_\infty$ threat model.

6. Conclusions

We consider the problem of hard label black-box adversarial attacks in low query budget regimes. To efficiently generate adversarial attacks with higher success rates and fewer queries, we define two low dimension structured subspace favorable for $\ell_2$ and $\ell_\infty$ norm constrained hard-label black box attacks. Our proposed method uses Bayesian optimization for finding adversarial perturbations in the low dimension subspace and maps it back to original input space to obtain final perturbation. We successfully demonstrate the efficacy of our method in attacking multiple deep learning architectures for high dimensional inputs in both $\ell_\infty$ and $\ell_2$ threat models. Our work opens avenues regarding applying BO for adversarial attacks in high dimensional settings.
References


7. Appendix

7.1. Untargeted $\ell_2$ attack

Figure 3 and 4 compare the performance across different $\ell_2$ perturbation bounds $\epsilon = 5.0$ and $\epsilon = 10.0$, and exhibits the relationship between success rates and number of queries used for each method. We can see that Bayes-Attack consistently outperforms the other baseline methods for across all query budgets up to 1000.

![Graphs showing performance comparison for $\ell_2$ untargeted attacks.](image)

*Figure 3. Performance comparison for $\ell_2$ untargeted attacks with $\epsilon = 10.0$ on ImageNet classifiers. Bayes-Attack significantly outperforms all the other baselines.*

7.2. Untargeted $\ell_\infty$ attack

Figure 5 compares the performance of the proposed method Bayes-Attack against Sign-OPT within the $\ell_\infty$ threat model with $\epsilon = 0.05$. The proposed method Bayes-Attack consistently achieve better performance for across all query budgets up to 1000.

7.3. Low Dimension Subspaces

Figure 6 compares the attacks generated in the low dimension subspace created using cosine and sine FFT basis vectors separately and together. We also compare with a random set of vectors sampled from standard normal distribution.
Figure 5. Performance comparison for $\ell_\infty$ untargeted attacks with $\epsilon = 0.05$ on ImageNet classifiers. Bayes-Attack significantly outperforms the other baseline.

Figure 6. Comparison of FFT basis vectors and random vectors sampled from the standard normal distribution for $\ell_2$ attack with $\epsilon = 20.0$ on ResNet50.