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

Satya Narayan Shukla [1] **Anit Kumar Sahu** [2] **Devin Willmott** [2] **Zico Kolter** [2 3]

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

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

Preliminary work.

case for Bayesian optimization. Rather, we show that we can bridge this gap by performing Bayesian optimization in a reduced-dimension setting by considering structured subspaces and mapping it back to full input space to obtain our final perturbation. We explore several mapping techniques and find that reducing the search space to a structured subspace composed of Fast Fourier Transform (FFT) basis vectors and a simple nearest-neighbor upsampling method allows us to sufficiently reduce the optimization problem dimension such that Bayesian optimization can find adversarial perturbations with more success than existing hard-label black-box attacks in query-constrained settings.

We compare the efficacy of our adversarial attack with a set of experiments attacking three of the most commonly used pretrained ImageNet (Deng et al., 2009) classifiers: ResNet50 (He et al., 2015), Inception-v3 (Szegedy et al., 2015), and VGG16-bn (Simonyan & Zisserman, 2014). We perform both ℓ_∞ and ℓ_2 norm constrained black-box attacks. Results from these experiments show that with small query budgets upto 1000, the proposed method BayesAttack achieves significantly better attack success rates than those of existing methods, and does so with far smaller average query counts.

Furthermore, ablation experiments are performed so as to compare the effectiveness of the configurations considered in our attack setup, i.e., selection of the structured low dimensional subspace and mapping techniques to generate adversarial perturbations in original image space. Given these results we argue that, despite being a simple approach (indeed, largely *because* it is such a simple and standard approach for black-box optimization), Bayesian Optimization should be a standard baseline for any hard-label black-box adversarial attack task in the future, especially in the small query budget regime.

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) 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

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 $F : \mathbb{R}^{d'} \rightarrow \{1, \dots K\}$ is a K-class image classifier that takes normalized RGB inputs: $\mathbf{x} \in \mathbb{R}^{d'}$ where each channel of each pixel is bounded between 0 and 1, $y \in \{1, \dots, K\}$ denotes the original label, and the corresponding output $F(\mathbf{x})$ 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 $\mathbf{x}$, the goal of the attacker is to find a perturbation $\boldsymbol{\delta}$ such that $\mathbf{x} + \boldsymbol{\delta}$ is misclassified, i.e., $F(\mathbf{x} + \boldsymbol{\delta}) \neq F(\mathbf{x})$. 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 $\|\boldsymbol{\delta}\|_p$ be smaller than a given threshold ϵ in some ℓ_p norm.In this work, we specifically focus on ℓ_2 and ℓ_∞ 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:

$$\max_{\boldsymbol{\delta}} \; f(\mathbf{x}, y, \boldsymbol{\delta}) \tag{1}$$
$$\text{subject to} \;\; \|\boldsymbol{\delta}\|_p \leq \epsilon \;\; \text{and} \;\; (\mathbf{x} + \boldsymbol{\delta}) \in [0, 1]^{d'},$$
$$\text{where} \;\; f(\mathbf{x}, y, \boldsymbol{\delta}) = \begin{cases} 0 & \text{if } F(\mathbf{x} + \boldsymbol{\delta}) \neq y \\ -1 & \text{if } F(\mathbf{x} + \boldsymbol{\delta}) = y \end{cases}$$

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

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 ℓ_∞ and ℓ_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 $\boldsymbol{\delta}$ in ℓ_1 norm instead of ℓ_2 and ℓ_∞ 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 μ_0 and Matern kernel (Shahriari et al., 2016; Snoek et al., 2012) as the covariance function Σ_0, which is defined as:

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

where d' is the dimension of input and $\{\theta_i\}_{i=0}^{d'}$ and μ_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 $\mathcal{A}$, 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 $\mathbf{x}_n = \arg\max_{\mathbf{x}} \mathcal{A}(\mathbf{x}|\mathcal{D}_{1:n-1})$ where $\mathcal{D}_{1:n-1}$ 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

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

where $\mathbb{E}_n[\cdot] = \mathbb{E}[\cdot|\mathcal{D}_{1:n-1}]$ denotes the expectation taken over the posterior distribution given evaluations of h at $\mathbf{x}_1, \cdots, \mathbf{x}_{n-1}$, 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 ℓ_2 and ℓ_∞ norm constrained hard label black-box attacks.

4.2.1. Low Dimensional Subspace for ℓ_2 norm constrained attack

To generate ℓ_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 $\mathbf{x} \in \mathbb{R}^{d \times d}$, the output of the FFT transform $\mathbf{X} := \text{FFT}(\mathbf{x})$ is defined by

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

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

$$\mathbf{x}[i, j] = \frac{1}{d} \sum_{u=0}^{d-1} \sum_{v=0}^{d-1} \mathbf{X}[u, v] \exp\left[j\frac{2\pi}{d}(u \cdot i + v \cdot j)\right] \tag{3}$$

The isometric property holds in reverse direction too, i.e. $\|\mathbf{X}\|_2 = \|\text{IFFT}(\mathbf{X})\|_2$. 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 $\mathbf{X}(u, v)$ respectively. To restrict to low-frequencies, we allow only elements in the top-left $\lfloor rd \rfloor \times \lfloor rd \rfloor$ square of $\mathbf{X}$ 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 $\mathbf{X}(u, v) = 0$ if $u > \lfloor rd \rfloor$ or $v > \lfloor rd \rfloor$. The adversarial perturbation is then obtained by computing $\text{IFFT}(\mathbf{X})$.

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 $\mathbf{X}$ 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.

Algorithm 1 Objective Function

procedure OBJ-FUNC($\mathbf{x}_0, y_0, \boldsymbol{\delta}$)
  // ϵ is the given perturbation
  $\boldsymbol{\delta} \leftarrow \Pi^p_{B(\mathbf{0},\epsilon)}(\boldsymbol{\delta})$ ▷ Projecting perturbation on ℓ_p-ball
  $\boldsymbol{\delta}' \leftarrow$ map($\boldsymbol{\delta}$) ▷ Mapping perturbation from low dimension subspace to full input space
  $v \leftarrow f(\mathbf{x}_0, y_0, \boldsymbol{\delta}')$ ▷ Quering the model
  return v

4.2.2. Low Resolution Subspace for ℓ_∞ norm constrained attack

Our method uses a data dependent prior (Ilyas et al., 2019) to reduce the search dimension of the perturbation for generating ℓ_∞ 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 $\mathbf{x} \in \mathbb{R}^{d\times d}$, we search for perturbations in a lower resolution image space $\mathbf{X} \in \mathbb{R}^{\lfloor rd \rfloor \times \lfloor rd \rfloor}$ where $r \in (0, 1]$ and use nearest neighbor interpolation (NNI) $\mathbf{x}' = \text{NNI}(\mathbf{X})$ to obtain the final adversarial perturbation. We note that $\mathbf{x}' \neq \mathbf{x}$. The NNI transformation leads to equivalent ℓ_∞ norms, i.e., $\|\mathbf{X}\|_\infty = \|\text{NNI}(\mathbf{X})\|_\infty$. 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 ($\lfloor rd \rfloor \times \lfloor rd \rfloor$) where $r \in (0, 1]$ and then use the relevant mapping (IFFT for ℓ_2 or and NNI for ℓ_∞) 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 $\Pi^p_{B(\mathbf{0},\epsilon)}$ be the projection onto the ℓ_p ball of radius ϵ 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 ℓ_p norm constraint by projecting the low dimension subspace perturbation on a ℓ_p ball of radius ϵ. Since, our mapping techniques (IFFT for ℓ_2 and NNI for ℓ_∞) do not change their respective norms, the final adversarial perturbation obtained after mapping to original input space also follows the ℓ_p constraint.

We describe the complete algorithm in Algorithm 2 where $\mathbf{x}_0 \in \mathbb{R}^{d\times d}$ and $y_0 \in \{1, \ldots, K\}$ denote the original input image[1] and label respectively. The goal is to learn an adversarial perturbation $\boldsymbol{\delta} \in \mathbb{R}^{\lfloor rd \rfloor \times \lfloor rd \rfloor}$ in much lower dimension, i.e., $r << 1$. We begin with a small dataset $\mathcal{D} = \{(\boldsymbol{\delta}_1, v_1), \cdots, (\boldsymbol{\delta}_{n_0}, v_{n_0})\}$ where each $\boldsymbol{\delta}_n$ is a $\lfloor rd \rfloor \times \lfloor rd \rfloor$ perturbation sampled from a given distribution and v_n is the function evaluation at $\boldsymbol{\delta}_n$ i.e $v_n =$ OBJ-FUNC$(\mathbf{x}_0, y_0, \boldsymbol{\delta}_n)$. 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 $\lfloor rd \rfloor \times \lfloor rd \rfloor$ 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 image, the low dimension subspace is of form $3 \times \lfloor rd \rfloor \times \lfloor rd \rfloor$ and the whole algorithm works the same way.

In this work, we focus on ℓ_∞ and ℓ_2-norm perturbations, where the respective projections are defined as:

$$\Pi^\infty_{B(\mathbf{x_0},\epsilon)}(\mathbf{x}) = \min\left\{\max\{\mathbf{x_0} - \epsilon, \mathbf{x}\}, \mathbf{x_0} + \epsilon\right\},$$
$$\Pi^2_{B(\mathbf{x_0},\epsilon)}(\mathbf{x}) = \underset{\|y - x_0\|_2 \leq \epsilon}{\arg\min} \|y - x\|_2$$

where ϵ is the given perturbation bound.

The initial choice of the dataset $\mathcal{D}$ 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 ℓ_p-constrained perturbation such that applying this perturbation to the original image causes misclassification. We evaluate both ℓ_∞ and ℓ_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

[1] For simplicity, we assume a 2D image here, our method can be easily applied to multi-channel images.

Algorithm 2 Adversarial Attack using Bayesian Optimization

procedure BAYES-ATTACK(x_0, y_0)
  $\mathcal{D} = \{(\boldsymbol{\delta}_1, v_1), \cdots, (\boldsymbol{\delta}_{n_0}, v_{n_0})\}$ ▷ Quering randomly chosen n_0 points.
  Update the GP on $\mathcal{D}$ ▷ Updating posterior distribution using available points
  $t \leftarrow n_0$ ▷ Updating number of queries till now
  while $t \leq T$ **do**
    $\boldsymbol{\delta}_t \leftarrow \arg\max_{\boldsymbol{\delta}} \mathcal{A}(\boldsymbol{\delta} \mid \mathcal{D})$ ▷ Optimizing the acquisition function over the GP
    $v_t \leftarrow$ OBJ-FUNC($\mathbf{x}_0, y_0, \boldsymbol{\delta}$) ▷ Querying the model
    $t \leftarrow t + 1$
    if $v_t < 0$ **then**
      $\mathcal{D} \leftarrow \mathcal{D} \cup (\boldsymbol{\delta}_t, v_t)$ and update the GP ▷ Updating posterior distribution
    else
      return $\boldsymbol{\delta}_t$ ▷ Adversarial attack successful
  return $\boldsymbol{\delta}_t$ ▷ Adversarial attack unsuccessful

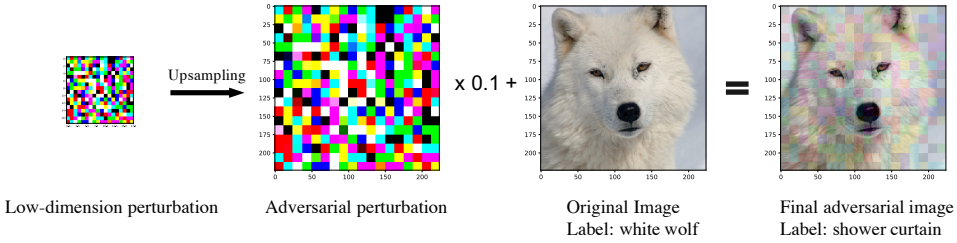


Figure 1. An illustration of a black-box adversarial attack performed by the proposed method BAYES-ATTACK on RESNET50 trained on ImageNet. Images from the left: first figure shows the learnt perturbation in low dimension $d' = 972(3 \times 18 \times 18)$; second figure is the final adversarial perturbation $(3 \times 224 \times 224)$ obtained by using nearest neighbor upsampling; third figure is the original image (note that the input size for RESNET50 is $3 \times 224 \times 224$) which is initially classified as *white/arctic wolf*; last image is the final adversarial image obtained by adding the adversarial perturbation to the original image. RESNET50 classifies the final adversarial image as *shower curtain* with high probability.

of Bayes-Attack on ImageNet classifiers with that of other hard-label black-box attacks for small query budgets, and report success rates and average queries. We also perform ablation studies on the ImageNet validation set by exploring different low dimension structured subspaces and examining different upsampling techniques.

We define success rate as the ratio of the number of images successfully perturbed for a given query budget to the total number of input images. In all experiments, images that are already misclassified by the target network are excluded from the set; only images that are initially classified with the correct label are attacked. For each method of attack and each target network, we compute the success rate and average number of queries used to attack among images that were successfully perturbed.

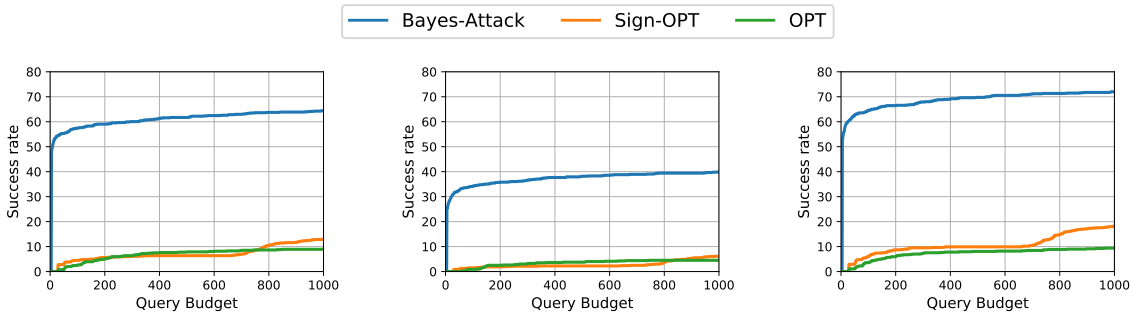
5.1. Empirical Protocols

We treat the size of low dimensional subspace used for running the Bayesian optimization loop as a hyperparameter. For both ℓ_2 and ℓ_∞ attacks on ImageNet, we tune the the low dimensional subspace size $\lfloor rd \rfloor \times \lfloor rd \rfloor$ over the range $rd \in [5, 18]$. For ℓ_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 ℓ_∞ attacks, and from the uniform distribution $[-1, 1]$ for ℓ_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 ℓ_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 ℓ_2 threat

[2] https://botorch.org/



(a) ResNet50 (b) Inception-v3 (c) VGG16-bn

Figure 2. Performance comparison for ℓ_2 untargeted attacks with $\epsilon = 20.0$ on ImageNet classifiers. Bayes-Attack significantly outperforms all the other baselines.

Table 1. Results for ℓ_2 untargeted attacks on ImageNet classifiers with a query budget of 1000

		$\epsilon = 5.0$		$\epsilon = 10.0$		$\epsilon = 20.0$	
Classifier	**Method**	**Success**	**Avg Queries**	**Success**	**Avg Queries**	**Success**	**Avg Queries**
ResNet50	OPT attack	2.16%	334.00	4.71%	294.62	9.03%	240.94
	Sign-OPT attack	4.07%	456.47	6.62%	424.62	12.98%	458.99
	Bayes attack	**20.10%**	**64.23**	**37.15%**	**64.13**	**66.67%**	**54.97**
Inception-v3	OPT attack	1.00%	277.75	1.88%	221.47	4.51%	247.67
	Sign-OPT attack	2.25%	648.50	3.75%	611.77	6.13%	557.71
	Bayes attack	**11.39%**	**109.65**	**22.65%**	**65.66**	**39.92%**	**68.86**
VGG16-bn	OPT attack	2.51%	204.47	4.10%	176.32	9.38%	229.77
	Sign-OPT attack	7.27%	552.33	11.36%	476.57	18.23%	417.46
	Bayes attack	**24.04%**	**69.84**	**43.46%**	**76.54**	**71.99%**	**48.95**

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 ℓ_2 perturbation bounds, where we set ϵ 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][5] 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 ϵ 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 ℓ_∞ 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 ℓ_∞ threat model. We use the same query budgets, models, and reported metrics as in the previous subsection. Here, we set the ℓ_∞ perturbation bound ϵ to 0.05. Table 2 compares the performance of ℓ_∞ 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/1ObXfAjAHm9FJfLy41TJfTlgQ4fSiP8Eq/edit

[5]https://github.com/LeMinhThong/blackbox-attack

Table 2. Results for ℓ_∞ untargeted attacks on ImageNet classifiers with a query budget of 1000

Classifier	Method	Success Rate	Avg Queries
ResNet50	Sign-OPT attack	3.05%	332.38
	Bayes attack	**67.48%**	**45.94**
Inception-v3	Sign-OPT attack	1.75%	549.00
	Bayes attack	**44.29%**	**72.31**
VGG16-bn	Sign-OPT attack	5.81%	469.84
	Bayes attack	**78.47%**	**33.70**

and comparing mapping techniques for both ℓ_2 and ℓ_∞ threat models.

5.4.1. Low Dimension Subspaces

In case of the ℓ_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 ℓ_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 almost same across the experiments, i.e. $3 \times 18 \times 18$ for cosine and sine FFT basis separately and $3 \times 12 \times 12 \times 2$ when considering the complete FFT basis. We also compare with a random set of vectors sampled from standard normal distribution. We keep the size of vectors sampled from normal distribution same as $3 \times 18 \times 18$.

Table 3. Performance comparison of FFT basis vectors and random vectors sampled from the standard normal distribution for ℓ_2 attack with $\epsilon = 20.0$ on ResNet50.

Basis	Success	Avg Queries
Cosine FFT	64.38%	54.25
Sine FFT	63.74%	45.72
Cosine and sine FFT	66.67%	54.97
Standard Normal	33.33%	48.25

Table 3 compares the performance of basis vectors in terms of attack success rate and average query count. We observe that the low frequency FFT basis vectors enhanced the attack accuracy significantly as compared to random set of vectors generated from standard normal distribution. On the other hand among low frequency components, sine and cosine FFT basis vectors together provide a slight gain in attack success rate as compared to using them separately.

5.4.2. Mapping Techniques

The proposed method requires a low dimensional subspace for efficiently searching the perturbation and a mapping technique for transforming the perturbation learnt in the low dimension to the full input space. We use FFT basis vectors for learning perturbation for ℓ_2 threat model while nearest neighbor interpolation for ℓ_∞ threat model. Here, we compare both the methods on ℓ_2 as well as ℓ_∞ threat models.

We compare both the mapping techniques on attacking ResNet50 trained on ImageNet with ℓ_∞ and ℓ_2 perturbation set to 0.05 and 20.0, respectively. 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 ℓ_∞ and ℓ_2 threat models. FFT basis vectors perform better than nearest neighbor interpolation in the ℓ_2 threat model, while nearest neighbor performs better for ℓ_∞ threat model.

Table 4. Comparing inverse fast Fourier transform (IFFT) and nearest neighbor interpolation (NNI) for ℓ_2 and ℓ_∞ attack on ResNet50.

Attack Type	Mapping Technique	Success Rate	Avg Queries
$\ell_\infty, \epsilon = 0.05$	IFFT	59.16%	55.72
	NNI	67.48%	45.94
$\ell_2, \epsilon = 20.0$	IFFT	66.67%	54.97
	NNI	59.54%	50.71

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 ℓ_2 and ℓ_∞ 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 ℓ_∞ and ℓ_2 threat models. Our work opens avenues regarding applying BO for adversarial attacks in high dimensional settings.

References

Brendel, W., Rauber, J., and Bethge, M. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint arXiv:1712.04248*, 2017.

Brunner, T., Diehl, F., Le, M. T., and Knoll, A. Guessing smart: Biased sampling for efficient black-box adversarial attacks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4958–4966, 2019.

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

Chen, J., Jordan, M. I., and Wainwright, M. J. Hopskipjumpattack: A query-efficient decision-based attack. *arXiv preprint arXiv:1904.02144*, 2019.

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 10th ACM Workshop on Artificial Intelligence and Security*, AISec '17, pp. 15–26, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-5202-4. doi: 10.1145/3128572.3140448.

Cheng, M., Le, T., Chen, P.-Y., Yi, J., Zhang, H., and Hsieh, C.-J. Query-efficient hard-label black-box attack: An optimization-based approach. *arXiv preprint arXiv:1807.04457*, 2018.

Cheng, M., Singh, S., Chen, P. H., Chen, P.-Y., Liu, S., and Hsieh, C.-J. Sign-{opt}: A query-efficient hard-label adversarial attack. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SklTQCNtvS.

Co, K. *Bayesian Optimization for Black-Box Evasion of Machine Learning Systems*. PhD thesis, Imperial College London, 2017.

Co, K. T., Muñoz-González, L., and Lupu, E. C. Procedural noise adversarial examples for black-box attacks on deep neural networks. *arXiv preprint arXiv:1810.00470*, 2018.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

Frazier, P. I. A tutorial on bayesian optimization. *ArXiv*, abs/1807.02811, 2018.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Gopakumar, S., Gupta, S., Rana, S., Nguyen, V., and Venkatesh, S. Algorithmic assurance: An active approach to algorithmic testing using bayesian optimisation. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31*, pp. 5465–5473. Curran Associates, Inc., 2018.

Guo, C., Frank, J. S., and Weinberger, K. Q. Low frequency adversarial perturbation. In *UAI*, 2018.

Guo, C., Gardner, J. R., You, Y., Wilson, A. G., and Weinberger, K. Q. Simple black-box adversarial attacks. *arXiv preprint arXiv:1905.07121*, 2019.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2015.

Ilyas, A., Engstrom, L., Athalye, A., and Lin, J. Black-box adversarial attacks with limited queries and information. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2137–2146, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

Ilyas, A., Engstrom, L., and Madry, A. Prior convictions: Black-box adversarial attacks with bandits and priors. In *International Conference on Learning Representations*, 2019.

Jones, D. R., Schonlau, M., and Welch, W. J. Efficient global optimization of expensive black-box functions. *J. of Global Optimization*, 13(4):455–492, December 1998. ISSN 0925-5001. doi: 10.1023/A:1008306431147.

Liu, Y., Chen, X., Liu, C., and Song, D. Delving into transferable adversarial examples and black-box attacks. *arXiv preprint arXiv:1611.02770*, 2016.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Moon, S., An, G., and Song, H. O. Parsimonious black-box adversarial attacks via efficient combinatorial optimization. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 4636–4645, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

Papernot, N., McDaniel, P. D., Goodfellow, I. J., Jha, S., Celik, Z. B., and Swami, A. Practical black-box attacks against machine learning. *CoRR*, abs/1602.02697, 2016.

Rasmussen, C. E. and Williams, C. K. I. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005. ISBN 026218253X.

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and de Freitas, N. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104: 148–175, 2016.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition, 2014. cite arxiv:1409.1556.

Snoek, J., Larochelle, H., and Adams, R. P. Practical bayesian optimization of machine learning algorithms. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 25*, pp. 2951–2959. Curran Associates, Inc., 2012.

Suya, F., Tian, Y., Evans, D., and Papotti, P. Query-limited black-box attacks to classifiers. *arXiv preprint arXiv:1712.08713*, 2017.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, 2015.

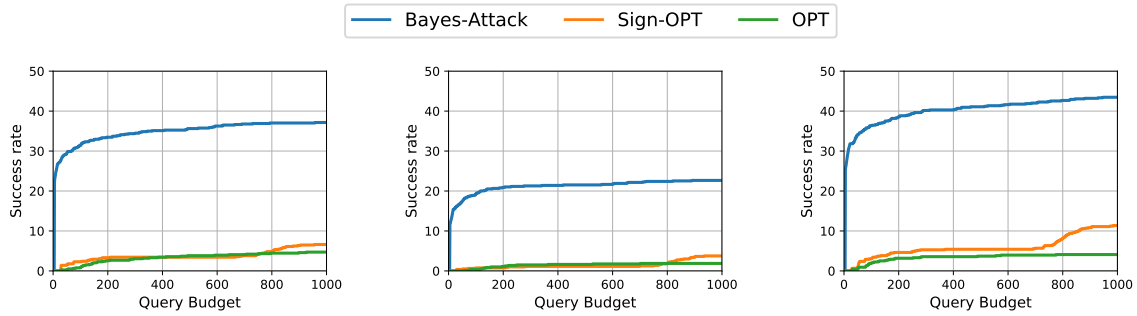
Tu, C., Ting, P., Chen, P., Liu, S., Zhang, H., Yi, J., Hsieh, C., and Cheng, S. Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019.*, pp. 742–749, 2019.

Zhao, P., Liu, S., Chen, P.-Y., Hoàng, N. N., Xu, K., Kailkhura, B., and Lin, X. C. On the design of black-box adversarial examples by leveraging gradient-free optimization and operator splitting method. *ArXiv*, abs/1907.11684, 2019.

7. Appendix

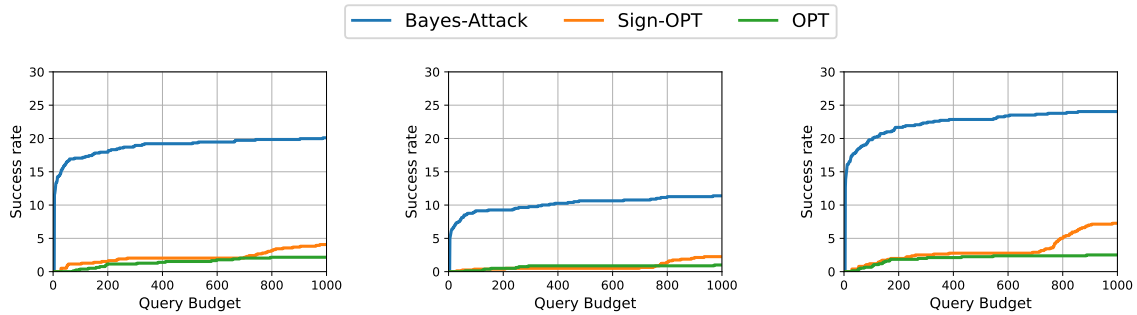
7.1. Untargeted ℓ_2 attack

Figure 3 and 4 compare the performance across different ℓ_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.



(a) ResNet50 (b) Inception-v3 (c) VGG16-bn

Figure 3. Performance comparison for ℓ_2 untargeted attacks with $\epsilon = 10.0$ on ImageNet classifiers. Bayes-Attack significantly outperforms all the other baselines.



(a) ResNet50 (b) Inception-v3 (c) VGG16-bn

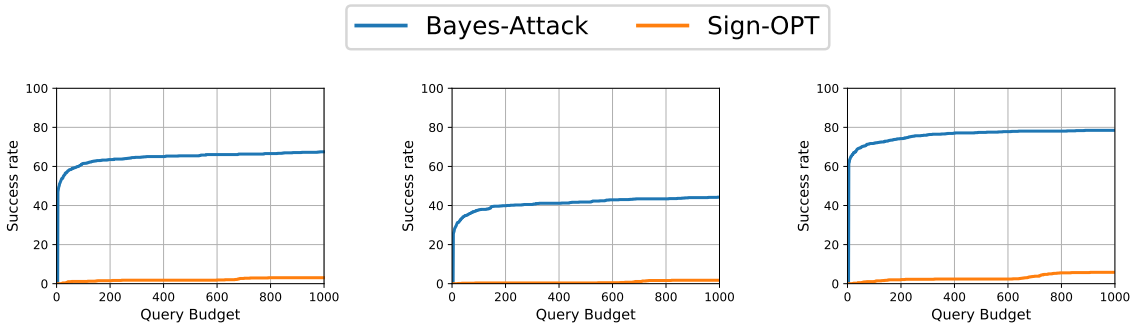
Figure 4. Performance comparison for ℓ_2 untargeted attacks with $\epsilon = 5.0$ on ImageNet classifiers. Bayes-Attack significantly outperforms all the other baselines.

7.2. Untargeted ℓ_∞ attack

Figure 5 compares the performance of the proposed method Bayes-Attack against Sign-OPT within the ℓ_∞ 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 Dimesion 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.



(a) ResNet50 (b) Inception-v3 (c) VGG16-bn

Figure 5. Performance comparison for ℓ_∞ untargeted attacks with $\epsilon = 0.05$ on ImageNet classifiers. Bayes-Attack significantly outperforms the other baseline.

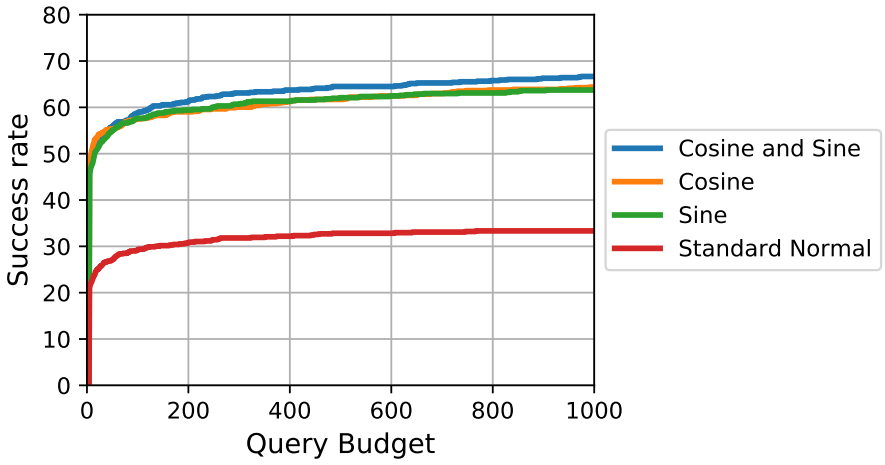


Figure 6. Comparison of FFT basis vectors and random vectors sampled from the standard normal distribution for ℓ_2 attack with $\epsilon = 20.0$ on ResNet50.