GeoDA: a geometric framework for black-box adversarial attacks

Ali Rahmati*, Seyed-Mohsen Moosavi-Dezfooli†, Pascal Frossard‡, and Huaiyu Dai*

*Department of ECE, North Carolina State University
†Institue for Machine Learning, ETH Zurich
‡Ecole Polytechnique Federale de Lausanne

arahmat@ncsu.edu, seyed.moosavi@inf.ethz.ch, pascal.frossard@epfl.ch, hdai@ncsu.edu

Abstract

Adversarial examples are known as carefully perturbed images fooling image classifiers. We propose a geometric framework to generate adversarial examples in one of the most challenging black-box settings where the adversary can only generate a small number of queries, each of them returning the top-1 label of the classifier. Our framework is based on the observation that the decision boundary of deep networks usually has a small mean curvature in the vicinity of data samples. We propose an effective iterative algorithm to generate query-efficient black-box perturbations with small ℓ_p norms for $p \geq 1$, which is confirmed via experimental evaluations on state-of-the-art natural image classifiers. Moreover, for $p = 2$, we theoretically show that our algorithm actually converges to the minimal ℓ_2-perturbation when the curvature of the decision boundary is bounded. We also obtain the optimal distribution of the queries over the iterations of the algorithm. Finally, experimental results confirm that our principled black-box attack algorithm performs better than state-of-the-art algorithms as it generates smaller perturbations with a reduced number of queries.[1]

1. Introduction

It has become well known that deep neural networks are vulnerable to small adversarial perturbations, which are carefully designed to cause miss-classification in state-of-the-art image classifiers [26]. Many methods have been proposed to evaluate adversarial robustness of classifiers in the white-box setting, where the adversary has full access to the target model [14, 24, 2]. However, the robustness of classifiers in black-box settings – where the adversary has only access to the output of the classifier – is of high relevance in many real-world applications of deep neural networks such as autonomous systems and healthcare, where it poses serious security threats. Several black-box evaluation methods have been proposed in the literature. Depending on what the classifier gives as an output, black-box evaluation methods are either score-based [25, 5, 18] or decision-based [3, 1, 20].

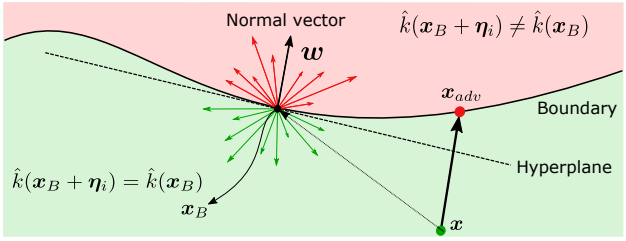


Figure 1: Linearization of the decision boundary.

In this paper, we propose a novel geometric framework for decision-based black-box attacks in which the adversary only has access to the *top-1 label* of the target model. Intuitively small adversarial perturbations should be searched in directions where the classifier decision boundary comes close to data samples. We exploit the low mean curvature of the decision boundary in the vicinity of the data samples to effectively estimate the normal vector to the decision boundary. This key prior permits to considerably reduces the number of queries that are necessary to fool the black-box classifier. Experimental results confirm that our Geometric Decision-based Attack (GeoDA) outperforms state-of-the-art black-box attacks, in terms of required number of queries to fool the classifier. Our main contributions are summarized as follows:

- We propose a novel geometric framework based on linearizing the decision boundary of deep networks in the vicinity of samples. The error for the estimation of the normal vector to the decision boundary of classifiers with flat decision boundaries, including linear classifiers, is shown to be bounded in a non-asymptotic regime. The proposed framework is general enough to be deployed for any classifier with low curvature decision boundary.

[1]The code of GeoDA is available at https://github.com/thisisalirah/GeoDA.

- We demonstrate how our proposed framework can be used to generate query-efficient ℓ_p black-box perturbations. In particular, we provide algorithms to generate perturbations for $p \geq 1$, and show their effectiveness via experimental evaluations on state-of-the-art natural image classifiers. In the case of $p = 2$, we also prove that our algorithm converges to the minimal ℓ_2-perturbation. We further derive the optimal number of queries for each step of the iterative search strategy.
- Finally, we show that our framework can incorporate different prior information, particularly transferability and subspace constraints on the adversarial perturbations. We show theoretically that having prior information can bias the normal vector estimation search space towards a more accurate estimation.

2. Related work

Adversarial examples can be crafted in white-box setting [14, 24, 2], score-based black-box setting [25, 5, 18] or decision-based black-box scenario [3, 1, 20]. The latter settings are obviously the most challenging as little is known about the target classification settings. Yet, there are several recent works on the black-box attacks on image classifiers [18, 19, 29]. However, they assume that the loss function, the prediction probabilities, or several top sorted labels are available, which may be unrealistic in many real-world scenarios. In the most challenging settings, there are a few attacks that exploit only the top-1 label information returned by the classifier, including the Boundary Attack (BA) [1], the HopSkipJump Attack (HSJA) [4], the OPT attack [7], and qFool [20]. In [1], by starting from a large adversarial perturbation, BA can iteratively reduce the norm of the perturbation. In [4], the authors provided an attack based on [1] that improves the BA taking the advantage of an estimated gradient. This attack is quite query efficient and can be assumed as the state-of-the-art baseline in the black-box setting. In [7], an optimization-based hard-label black-box attack algorithm is introduced with guaranteed convergence rate in the hard-label black-box setting which outperforms the BA in terms of number of queries. Closer to our work, in [20], a heuristic algorithm based on the estimation of the normal vector to decision boundary is proposed for the case of ℓ_2-norm perturbations.

Most of the aforementioned attacks are however specifically designed for minimizing perturbation metrics such ℓ_2 and ℓ_∞ norms, and mostly use heuristics. In contrast, we introduce a powerful and generic framework grounded on the geometric properties of the decision boundary of deep networks, and propose a principled approach to design efficient algorithms to generate general ℓ_p-norm perturbations, in which [20] can be seen as a special case. We also provide convergence guarantees for the ℓ_2-norm perturbations. We obtained the optimal distribution of queries over iterations theoretically as well which permits to use the queries in a more efficient manner. Moreover, the parameters of our algorithm are further determined via empirical and theoretical analysis, not merely based on heuristics as done in [20].

3. Problem statement

Let us assume that we have a pre-trained L-class classifier with parameters θ represented as $f : \mathbb{R}^d \to \mathbb{R}^L$, where $\boldsymbol{x} \in \mathbb{R}^d$ is the input image and $\hat{k}(\boldsymbol{x}) = \text{argmax}_k \; f_k(\boldsymbol{x})$ is the top-1 classification label where $f_k(\boldsymbol{x})$ is the k^{th} component of $f(\boldsymbol{x})$ corresponds to the k^{th} class. We consider the non-targeted black-box attack, where an adversary without any knowledge on θ computes an adversarial perturbation $\boldsymbol{v}$ to change the estimated label of an image $\boldsymbol{x}$ to any incorrect label, i.e., $\hat{k}(\boldsymbol{x} + \boldsymbol{v}) \neq \hat{k}(\boldsymbol{x})$. The distance metric $\mathcal{D}(\boldsymbol{x}, \boldsymbol{x} + \boldsymbol{v})$ can be any function including the ℓ_p norms. We formulate an optimization problem with the goal to fool the classifier while $\mathcal{D}(\boldsymbol{x}, \boldsymbol{x} + \boldsymbol{v})$ is minimized as:

$$\begin{aligned} \min_{\boldsymbol{v}} \quad & \mathcal{D}(\boldsymbol{x}, \boldsymbol{x} + \boldsymbol{v}) \\ \text{s.t.} \quad & \hat{k}(\boldsymbol{x} + \boldsymbol{v}) \neq \hat{k}(\boldsymbol{x}). \end{aligned} \tag{1}$$

Finding a solution for (1) is a hard problem in general. To obtain an efficient approximate solution, one can try to estimate the point of the classifier decision boundary that is the closest to the data point $\boldsymbol{x}$. Crafting an small adversarial perturbation then consists in pushing the data point beyond the decision boundary in the direction of its normal. The normal to the decision boundary is thus critical in a geometry-based attack. While it can be obtained using back-propagation in white box settings (e.g., [24]), its estimation in black-box settings becomes challenging.

The key idea here is to exploit the geometric properties of the decision boundary in deep networks for effective estimation in black-box settings. In particular, it has been shown that the decision boundaries of the state-of-the-art deep networks have a quite low mean curvature in the neighborhood of data samples [11]. Specifically, the decision boundary at the vicinity of a data point $\boldsymbol{x}$ can be locally approximated by a hyperplane passing through a boundary point $\boldsymbol{x}_B$ close to $\boldsymbol{x}$, with a normal vector $\boldsymbol{w}$ [13, 12]. Thus, by exploiting this property, the optimization problem in (1) can be locally linearized as:

$$\begin{aligned} \min_{\boldsymbol{v}} \quad & \mathcal{D}(\boldsymbol{x}, \boldsymbol{x} + \boldsymbol{v}) \\ \text{s.t.} \quad & \boldsymbol{w}^T(\boldsymbol{x} + \boldsymbol{v}) - \boldsymbol{w}^T\boldsymbol{x}_B = 0 \end{aligned} \tag{2}$$

Typically, $\boldsymbol{x}_B$ is a point on the boundary, which can be found by binary search with a small number of queries. However, solving the problem (2) is quite challenging in black-box settings as one does not have any knowledge about the parameters θ and can only access the top-1 label $\hat{k}(\boldsymbol{x})$ of the image classifier. A *query* is a request that results in the top-1 label of an image classifier for a given input, which prevents the use of zero-order black box optimization methods [31, 30] that need more information to

compute adversarial perturbations. The goal of our method is to estimate the normal vector to the decision boundary $\boldsymbol{w}$ resorting to geometric priors with a minimal number of queries to the classifier.

4. The estimator

We introduce an estimation method for the normal vector of classifiers with flat decision boundaries. It is worth noting that the proposed estimation is not limited to deep networks and applies to any classifier with low mean curvature boundary. We denote the estimate of the vector $\boldsymbol{w}$ normal to the flat decision boundary in (2) with $\hat{\boldsymbol{w}}_N$ when N queries are used. Without loss of generality, we assume that the boundary point $\boldsymbol{x}_B$ is located at the origin. Thus, according to (2), the decision boundary hyperplane passes through the origin and we have $\boldsymbol{w}^T\boldsymbol{x} = 0$ for any vector $\boldsymbol{x}$ on the decision boundary hyperplane. In order to estimate the normal vector to the decision boundary, the key idea is to generate N samples $\boldsymbol{\eta}_i$, $i \in \{1, \dots, N\}$ from a multivariate normal distribution $\boldsymbol{\eta}_i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$. Then, we query the image classifier N times to obtain the top-1 label output for each $\boldsymbol{x}_B + \boldsymbol{\eta}_i$, $\forall i \in N$. For a given data point $\boldsymbol{x}$, if $\boldsymbol{w}^T\boldsymbol{x} \leq 0$, the label is correct; if $\boldsymbol{w}^T\boldsymbol{x} \geq 0$, the classifier is fooled. Hence, if the generated perturbations are adversarial, they belong to the set

$$\begin{aligned} \mathcal{S}_{\text{adv}} &= \{\boldsymbol{\eta}_i \mid \hat{k}(\boldsymbol{x}_B + \boldsymbol{\eta}_i) \neq \hat{k}(\boldsymbol{x})\} \\ &= \{\boldsymbol{\eta}_i \mid \boldsymbol{w}^T\boldsymbol{\eta}_i \geq 0\}. \end{aligned} \tag{3}$$

Similarly, the perturbations on the other side of the hyperplane, which lead to correct classification, belong to the set

$$\begin{aligned} \mathcal{S}_{\text{clean}} &= \{\boldsymbol{\eta}_i \mid \hat{k}(\boldsymbol{x}_B + \boldsymbol{\eta}_i) = \hat{k}(\boldsymbol{x})\} \\ &= \{\boldsymbol{\eta}_i \mid \boldsymbol{w}^T\boldsymbol{\eta}_i \leq 0\}. \end{aligned} \tag{4}$$

The samples in each of the sets $\mathcal{S}_{\text{adv}}$ and $\mathcal{S}_{\text{clean}}$ can be assumed as samples drawn from a hyperplane ($\boldsymbol{w}^T\boldsymbol{x} = 0$) truncated multivariate normal distribution with mean $\mathbf{0}$ and covariance matrix $\boldsymbol{\Sigma}$. We define the PDF of the d dimensional zero mean multivariate normal distribution with covariance matrix $\boldsymbol{\Sigma}$ as $\phi_d(\boldsymbol{\eta}|\boldsymbol{\Sigma})$. We define $\Phi_d(\boldsymbol{b}|\boldsymbol{\Sigma}) = \int_{\boldsymbol{b}}^{\infty} \phi_d(\boldsymbol{\eta}|\boldsymbol{\Sigma}) d\boldsymbol{\eta}$ as cumulative distribution function of the univariate normal distribution.

Lemma 1. *Given a multivariate Gaussian distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ truncated by the hyperplane $\boldsymbol{w}^T\boldsymbol{x} \geq 0$, the mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{R}$ of the hyperplane truncated distribution are given by:*

$$\boldsymbol{\mu} = c_1 \boldsymbol{\Sigma} \boldsymbol{w} \tag{5}$$

where $c_1 = (\Phi_d(0))^{-1}\phi_d(0)$ and the covariance matrix $\boldsymbol{R} = \boldsymbol{\Sigma} - \boldsymbol{\Sigma}\boldsymbol{w}\boldsymbol{w}^T\boldsymbol{\Sigma}(\Phi_d(0)^2\gamma^2)^{-1}\phi_d(0))d^2(0)$ in which $\gamma = (\boldsymbol{w}^T\boldsymbol{\Sigma}\boldsymbol{w})^{\frac{1}{2}}$ [27].

As it can be seen in (5), the mean is a function of both the covariance matrix $\boldsymbol{\Sigma}$ and $\boldsymbol{w}$. Our ultimate goal is to estimate the normal vector to the decision boundary. In order to recover $\boldsymbol{w}$ from $\boldsymbol{\mu}$, a sufficient condition is to choose $\boldsymbol{\Sigma}$ to be a full rank matrix.

General case We first consider the case where no prior information on the search space is available. The matrix $\boldsymbol{\Sigma} = \sigma\mathcal{I}$ can be a simple choice to avoid unnecessary computations. The direction of the mean of the truncated distribution is an estimation for the direction of hyperplane normal vector as $\boldsymbol{\mu} = c_1\sigma\boldsymbol{w}$. The covariance matrix of the truncated distribution is $\boldsymbol{R} = \sigma\mathcal{I} + c_2\boldsymbol{w}\boldsymbol{w}^T$ where $c_2 = -\sigma^2(\Phi_d(0))^{-2}\phi_d^2(0)$. As the samples in both of the sets $\mathcal{S}_{\text{adv}}$ and $\mathcal{S}_{\text{clean}}$ are hyperplane truncated Gaussian distributions, the same estimation can be applied for the samples in the set $\mathcal{S}_{\text{clean}}$ as well. Thus, by multiplying the samples in $\mathcal{S}_{\text{clean}}$ by -1 and we can use them to approximate the desired gradient to have a more efficient estimation. Hence, the problem is reduced to the estimation of the mean of the N samples drawn from the hyperplane truncated distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{R}$. As a result, the estimator $\bar{\boldsymbol{\mu}}_N$ of $\boldsymbol{\mu}$ with N samples is $\bar{\boldsymbol{\mu}}_N = \frac{1}{N}\sum_{i=1}^{N} \rho_i\boldsymbol{\eta}_i$, where

$$\rho_i = \begin{cases} 1 & \hat{k}(\boldsymbol{x}_B + \boldsymbol{\eta}_i) \neq \hat{k}(\boldsymbol{x}) \\ -1 & \hat{k}(\boldsymbol{x}_B + \boldsymbol{\eta}_i) = \hat{k}(\boldsymbol{x}). \end{cases} \tag{6}$$

The normalized direction of the normal vector of the boundary can be obtained as:

$$\hat{\boldsymbol{w}}_N = \frac{\bar{\boldsymbol{\mu}}_N}{\|\bar{\boldsymbol{\mu}}_N\|_2} \tag{7}$$

Perturbation priors We now consider the case where priors on the perturbations are available. In black-box settings, having prior information can significantly improve the performance of the attack. Although the attacker does not have access to the weights of the classifier, it may have some prior information about the data, classifier, etc. [19]. Using $\boldsymbol{\Sigma}$, we can capture the prior knowledge for the estimation of the normal vector to the decision boundary. In the following, we unify two common priors in our proposed estimator.

In the first case, we have some prior information about the subspace in which we search for normal vectors, we can incorporate such information into $\boldsymbol{\Sigma}$ to have a more efficient estimation. For instance, deploying low frequency subspace $\mathcal{R}^m$ in which $m \ll d$, we can generate a rank m covariance matrix $\boldsymbol{\Sigma}$. Let us assume that $\mathcal{S} = \{\boldsymbol{s}_1, \boldsymbol{s}_2, ..., \boldsymbol{s}_m\}$ is an orthonormal Discrete Cosine Transform (DCT) basis in the m-dimensional subspace of the input space [15]. In order to generate the samples from this low dimensional subspace, we use the following covariance matrix:

$$\boldsymbol{\Sigma} = \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{s}_i\boldsymbol{s}_i^T. \tag{8}$$

The normal vector of the boundary can be obtained by plugging the modified $\mathbf{\Sigma}$ in (5).

Second, we consider transferability priors. It has been observed that adversarial perturbations well transfer across different trained models [28, 23, 8]. Now, if the adversary further has full access to another model $\mathcal{T}'$, yet different than the target black-box model $\mathcal{T}$, it can take advantage of the transferability properties of adversarial perturbations. For a given datapoint, one can obtain the normal vector to the decision boundary in the vicinity of the datapoint for $\mathcal{T}'$, and bias the normal vector search space for the black-box classifier. Let us denote the transferred direction with unit-norm vector $\boldsymbol{g}$. By incorporating this vector into $\mathbf{\Sigma}$, we can bias the search space as:

$$\mathbf{\Sigma} = \beta\mathcal{I} + (1-\beta)\boldsymbol{g}\boldsymbol{g}^T \tag{9}$$

where $\beta \in [0,1]$ adjusts the trade-off between exploitation and exploration. Depending on how confident we are about the utility of the transferred direction, we can adjust its contribution by tuning the value of β. Substituting (9) into (5), after normalization to c_1, one can get

$$\boldsymbol{\mu} = \beta\boldsymbol{w} + (1-\beta)\boldsymbol{g}\boldsymbol{g}^T\boldsymbol{w}, \tag{10}$$

where the first term is the estimated normal vector to the boundary and the second term is the projection of the estimated normal vector on the transferred direction $\boldsymbol{g}$. Having incorporated the prior information into $\mathbf{\Sigma}$, one can generate perturbations $\boldsymbol{\eta}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ with the modified $\mathbf{\Sigma}$ in an effective search space, which leads to a more accurate estimation of normal to the decision boundary.

Estimator bound Finally, we are interested in quantifying the number of samples that are necessary for estimating the normal vectors in our geometry inspired framework. Given a real i.i.d. sequence, using the central limit theorem, if the samples have a finite variance, an asymptotic bound can be provided for the estimate. However, this bound is not of our interest as it is only asymptotically correct. We are interested in bounds of similar form with non-asymptotic inequalities as the number of queries is limited [21, 16].

Lemma 2. *The mean estimation $\bar{\boldsymbol{\mu}}_N$ deployed in (9) obtained from N multivariate hyperplane truncated Gaussian queries satisfies the probability*

$$P\left(\|\bar{\boldsymbol{\mu}}_N - \boldsymbol{\mu}\| \leq \sqrt{\frac{Tr(\boldsymbol{R})}{N}} + \sqrt{\frac{2\lambda_{max}\log(1/\delta)}{N}}\right) \geq 1-\delta \tag{11}$$

where $Tr(\boldsymbol{R})$ and λ_{max} denote the trace and largest eigenvalue of the covariance matrix $\boldsymbol{R}$, respectively.

Proof. The proof can be found in Appendix A. □

This bound will be deployed in sub-section 5.1 to compute the optimal distribution of queries over iterations.

Algorithm 1: ℓ_p GeoDA (with optimal query distribution) for $p > 1$

Inputs: Original image $\boldsymbol{x}$, query budget N, λ, number of iterations T.
Output: Adversarial example $\boldsymbol{x}_T$.
Obtain the optimal query distribution $N_t^*, \forall t$ by (19).
Find a starting point on the boundary $\boldsymbol{x}_0$.
for $t = 1 : T$ **do**
  Estimate normal $\hat{\boldsymbol{w}}_{N_t^*}$ at $\boldsymbol{x}_{t-1}$ by N_t^* queries.
  Obtain $\boldsymbol{v}_t$ according to (13).
  $\hat{r}_t \leftarrow \min\{r' > 0 \ : \ \hat{k}(\boldsymbol{x} + r'\boldsymbol{v}_t) \neq \hat{k}(\boldsymbol{x})\}$
  $\boldsymbol{x}_t \leftarrow \boldsymbol{x} + \hat{r}_t\hat{\boldsymbol{w}}_{N_t^*}$

5. Geometric decision-based attacks (GeoDA)

Based on the estimator provided in Section 4, one can design efficient black-box evaluation methods. In this paper, we focus on the minimal ℓ_p-norm perturbations, i.e., $\mathcal{D}(\boldsymbol{x}, \boldsymbol{x}+\boldsymbol{v}) = \|\boldsymbol{v}\|_p$. We first describe the general algorithm for ℓ_p perturbations, and then provide algorithms to find black-box perturbations for $p = 1, 2, \infty$. Furthermore, for $p = 2$, we prove the convergence of our method. The linearized optimization problem in (2) can be re-written as

$$\begin{aligned} \min_{\boldsymbol{v}} \quad & \|\boldsymbol{v}\|_p \\ \text{s.t.} \quad & \boldsymbol{w}^T(\boldsymbol{x}+\boldsymbol{v}) - \boldsymbol{w}^T\boldsymbol{x}_B = 0. \end{aligned} \tag{12}$$

In the black-box setting, one needs to estimate $\boldsymbol{x}_B$ and $\boldsymbol{w}$ in order to solve this optimization problem. The boundary point $\boldsymbol{x}_B$ can be found using a similar approach as [20]. Having $\boldsymbol{x}_B$, one then use the process described in Section 4 to compute the estimator of $\boldsymbol{w}$ – i.e.,$\hat{\boldsymbol{w}}_{N_1}$ – by making N_1 queries to the classifier. In the case of $p = 2$, the estimated direction $\hat{\boldsymbol{w}}_N$ is indeed the direction of the minimal perturbation. This process is depicted in Fig. 1.

If the curvature of the decision boundary is exactly zero, the solution of this problem gives the direction of the minimal ℓ_p perturbation. However, for deep neural networks, even if $N \to \infty$, the obtained direction is not completely aligned with the minimal perturbation as these networks still have a small yet non-zero curvature (see Fig. 4c). Nevertheless, to overcome this issue, the solution $\boldsymbol{v}^*$ of (12) can be used to obtain a boundary point $\boldsymbol{x}_1 = \boldsymbol{x} + \hat{r}_1\boldsymbol{v}^*$ to the original image $\boldsymbol{x}$ than $\boldsymbol{x}_0$, for an appropriate value of $\hat{r}_1 > 0$. For notation consistency, we define $\boldsymbol{x}_0 = \boldsymbol{x}_B$. Now, we can again solve (12) for the new boundary point $\boldsymbol{x}_1$. Repeating this process results in an iterative algorithm to find the minimal ℓ_p perturbation, where each iteration corresponds to solving (12) once. Formally, for a given image $\boldsymbol{x}$, let $\boldsymbol{x}_t$ be the boundary point estimated in the iteration $t-1$. Also, let N_t be the number of queries used to estimate

the normal to the decision boundary $\hat{\boldsymbol{w}}_{N_t}$ at the iteration t. Hence, the (normalized) solution to (12) in the t-th iteration, $\boldsymbol{v}_t$, can be written in closed-form as:

$$\boldsymbol{v}_t = \frac{1}{\|\hat{\boldsymbol{w}}_{N_t}\|_{\frac{p}{p-1}}} \odot \text{sign}(\hat{\boldsymbol{w}}_{N_t}), \tag{13}$$

for $p \in [1, \infty)$, where $\odot$ is the point-wise product. For the particular case of $p = \infty$, the solution of (13) is simply reduced to:

$$\boldsymbol{v}_t = \text{sign}(\hat{\boldsymbol{w}}_{N_t}). \tag{14}$$

The cases of the $p = 1, 2$ are presented later. In all cases, $\boldsymbol{x}_t$ is then updated according to the following update rule:

$$\boldsymbol{x}_t = \boldsymbol{x} + \hat{r}_t \boldsymbol{v}_t \tag{15}$$

where $\hat{r}_t$ can be found using an efficient line search along $\boldsymbol{v}_t$. The general algorithm is summarized in Alg. 1.

5.1. ℓ_2 perturbation

In the ℓ_2 case, the update rule of (15) is reduced to $\boldsymbol{x}_t = \boldsymbol{x} + \hat{r}_t \hat{\boldsymbol{w}}_{N_t}$ where $\hat{r}_t$ is the ℓ_2 distance of $\boldsymbol{x}$ to the decision boundary at iteration t. We propose convergence guarantees and optimal distribution of queries over the successive iterations for this case.

Convergence guarantees We prove that GeoDA converges to the minimal ℓ_2 perturbation given that the curvature of the decision boundary is bounded. We define the curvature of the decision boundary as $\kappa = \frac{1}{R}$, where R is the radius of the largest open ball included in the region that intersects with the boundary $\mathcal{B}$ [11]. In case $N \to \infty$, then $\hat{r}_t \to r_t$ where r_t is assumed as exact distance required to push the image $\boldsymbol{x}$ towards the boundary at iteration t with direction $\boldsymbol{v}_t$. The following Theorem holds:

Theorem 1. *Given a classifier with decision boundary of bounded curvature with $\kappa r < 1$, the sequence $\{\hat{r}_t\}$ generated by Algorithm 1 converges linearly to the minimum ℓ_2 distance r since we have:*

$$\lim_{t \to \infty} \frac{\hat{r}_{t+1} - r}{\hat{r}_t - r} = \lambda \tag{16}$$

where $\lambda < 1$ is the convergence rate.

Proof. The proof can be found in Appendix B. □

Optimal query distribution In practice, however, the number of queries N is limited. One natural question is how should one choose the number of queries in each iteration of GeoDA. It can be seen in the experiments that allocating a smaller number of queries for the first iterations and then increasing it in each iteration can improve the convergence rate of the GeoDA. At early iterations, noisy normal vector estimates are fine because the noise is smaller relative to the potential improvement, whereas in later iterations noise has a bigger impact. This makes the earlier iterations cheaper in terms of queries, potentially speeding up convergence [10].

We assume a practical setting in which we have a limited budget N for the number of queries as the target system may block if the number of queries increases beyond a certain threshold [6]. The goal is to obtain the optimal distribution of the queries over the iterations.

Theorem 2. *Given a limited query budget N, the bounds for the GeoDA ℓ_2 perturbation error for total number of iterations T can be obtained as:*

$$\lambda^T (r_0 - r) - e(\boldsymbol{N}) \leq \hat{r}_t - r \leq \lambda^T (r_0 - r) + e(\boldsymbol{N}) \tag{17}$$

where $e(\boldsymbol{N}) = \gamma \sum_{i=1}^{T} \frac{\lambda^{T-i} r_i}{\sqrt{N_i}}$ is the error due to limited number of queries, $\gamma = \sqrt{Tr(\boldsymbol{R})} + \sqrt{2\lambda_{\max} \log(1/\delta)}$ and N_t is the number of queries to estimate the normal vector to the boundary at point $\boldsymbol{x}_{t-1}$, and $r_0 = \|\boldsymbol{x} - \boldsymbol{x}_0\|$.

Proof. The proof can be found in Appendix C. □

As in (17), the error in the convergence is due to two factors: (*i*) curvature of the decision boundary (*ii*) limited number of queries. If the number of iterations increases, the effect of the curvature can vanish. However, the term $\gamma \frac{r_i}{\sqrt{N_t}}$ is not small enough as the number of queries is finite. Having unlimited number of the queries, the error term due to queries can vanish as well. However, given a limited number of queries, what should be the distribution of the queries to alleviate such an error? We define the following optimization problem:

$$\begin{aligned} \min_{N_1, \ldots, N_T} \quad & \sum_{i=1}^{T} \frac{\lambda^{-i} r_i}{\sqrt{N_i}} \\ \text{s.t.} \quad & \sum_{i=1}^{T} N_i \leq N \end{aligned} \tag{18}$$

where the objective is to minimize the error $e(\boldsymbol{N})$ while the query budget constraint is met over all iterations.

Theorem 3. *The optimal numbers of queries for (18) in each iteration form geometric sequence with the common ratio $\frac{N^*_{t+1}}{N^*_t} \approx \lambda^{-\frac{2}{3}}$, where $0 \leq \lambda \leq 1$. Moreover, we have*

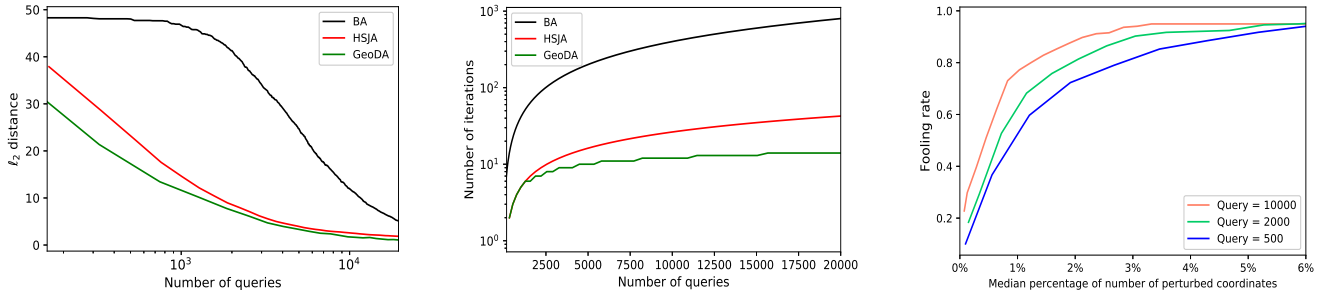
$$N^*_t \approx \frac{\lambda^{-\frac{2}{3}t}}{\sum_{i=1}^{T} \lambda^{-\frac{2}{3}i}} N. \tag{19}$$

Proof. The proof can be found in Appendix D. □

5.2. ℓ_1 perturbation (sparse case)

The framework proposed by GeoDA is general enough to find sparse adversarial perturbations in the black-box setting as well. The sparse adversarial perturbations can be computed using the following optimization problem with box constraints as:

$$\begin{aligned} \min_{\boldsymbol{v}} \quad & \|\boldsymbol{v}\|_1 \\ \text{s.t.} \quad & \boldsymbol{w}^T (\boldsymbol{x} + \boldsymbol{v}) - \boldsymbol{w}^T \boldsymbol{x}_B = 0 \\ & \boldsymbol{l} \preceq \boldsymbol{x} + \boldsymbol{v} \preceq \boldsymbol{u} \end{aligned} \tag{20}$$



(a) (b) (c)

Figure 2: Performance evaluation of GeoDA for ℓ_p when $p = 1, 2$ (a) Comparison for the performance of GeoDA, BA, and HSJA for ℓ_2 norm. (b) Comparison for the number of required iterations in GeoDA, BA, and HSJA. (c) Fooling rate vs. sparsity for different numbers of queries in sparse GeoDA.

In the box constraint $\boldsymbol{l} \preceq \boldsymbol{x} + \boldsymbol{v} \preceq \boldsymbol{u}$, $\boldsymbol{l}$ and $\boldsymbol{u}$ denote the lower and upper bounds of the values of $\boldsymbol{x} + \boldsymbol{v}$. We can estimate the normal vector $\hat{\boldsymbol{w}}_N$ and the boundary point $\boldsymbol{x}_B$ similarly to the ℓ_2 case with N queries. Now, the decision boundary $\mathcal{B}$ is approximated with the hyperplane $\{\boldsymbol{x} : \hat{\boldsymbol{w}}_N^T(\boldsymbol{x} - \boldsymbol{x}_B) = 0\}$. The goal is to find the top-k coordinates of the normal vector $\hat{\boldsymbol{w}}_N$ with minimum k and pushing them to extreme values of the valid range depending on the sign of the coordinate until it hits the approximated hyperplane. In order to find the minimum k, we deploy binary search for a d-dimensional image. Here, we just consider one iteration for the sparse attack., while the initial point of the sparse case is obtained using the GeoDA for ℓ_2 case. The detailed Algorithm for the sparse version of GeoDA is given in supplementary material.

6. Experiments

We evaluate our algorithms on a pre-trained ResNet-50 [17] with a set $\mathcal{X}$ of 350 correctly classified and randomly selected images from the ILSVRC2012's validation set [9]. All the images are resized to $224 \times 224 \times 3$.

To evaluate the performance of the attack we deploy the median of the ℓ_p norm for $p = 2, \infty$ distance over all tested samples, defined by $\underset{\boldsymbol{x} \in \mathcal{X}}{\text{median}} \Big(\|\boldsymbol{x} - \boldsymbol{x}^{\text{adv}}\|_p \Big)$. For sparse perturbations, we measure the performance by fooling rate defined as $|\boldsymbol{x} \in \mathcal{X} : \hat{k}(\boldsymbol{x}) \neq \hat{k}(\boldsymbol{x}^{\text{adv}})|/|\mathcal{X}|$. In evaluation of the sparse GeoDA, we define *sparsity* as the percentage of the perturbed coordinates of the given image.

6.1. Performance analysis

Black-box attacks for ℓ_p norms. We compare the performance of the GeoDA with state of the art attacks for ℓ_p norms. There are several attacks in the literature including Boundary attack [1], HopSkipJump attack [4], qFool [20], and OPT attack [7]. In our experiments, we compare GeoDA with Boundary attack, qFool and HopSkipJump attack. We do not compare our algorithm with OPT attack as HopSkipJump already outperforms it considerably [4]. In our algorithm, the optimal distribution of the queries is obtained for any given number of queries for ℓ_2 case. The results for ℓ_2 and ℓ_∞ for different numbers of queries is depicted in Table 2. GeoDA can outperform the-state-of-the-art both in terms of smaller perturbations and number of iterations, which has the benefit of parallelization. In particular, the images can be fed into multiple GPUs with larger batch size. In Fig. 2a, the ℓ_2 norm of GeoDA, Boundary attack and HopSkipJump are compared. As shown, GeoDA can outperform the HopSkipJump attack especially when the number of queries is small. By increasing the number of queries, the performance of GeoDA and HopSkipJump are getting closer.

	Queries	**Fooling rate**	**Perturbation**
GeoDA	500	88.44 %	4.29 %
	2000	90.25 %	3.04 %
	10000	91.17 %	2.36 %
SparseFool [1]	-	100 %	0.23 %

Table 1: The performance comparison of black-box sparse GeoDA for median sparsity compared to white box attack SparseFool [1] on ImageNet dataset.

In Fig. 2b, the number of iterations versus the number of queries for different algorithms are compared. As depicted, GeoDA needs fewer iterations compared to HopSkipJump and BA when the number of queries increases. Thus, on the one hand GeoDA generates smaller ℓ_2 perturbations compared to the HopSkipJump attack when the number of queries is small, on the other hand, it saves significant computation time due to parallelization.

Now, we evaluate the performance of GeoDA for generating sparse perturbations. In Fig. 2c, the fooling rate versus sparsity is depicted. In experiments, we observed that instead of using the boundary point $\boldsymbol{x}_B$ in the sparse GeoDA, the performance of the algorithm can be improved by further moving towards the other side of the hyperplane boundary. Thus, we use $\boldsymbol{x}_B + \zeta(\boldsymbol{x}_B - \boldsymbol{x})$, where $\zeta \geq 0$. The parameter ζ can adjust the trade-off between the fooling rate and the sparsity. It is observed that the higher the

	Queries	ℓ_2	ℓ_∞	Iterations	Gradients
Boundary attack [1]	1000	47.92	0.297	40	-
	5000	24.67	0.185	200	-
	20000	5.13	0.052	800	-
qFool [4]	1000	16.05	-	3	-
	5000	7.52	-	3	-
	20000	1.12	-	3	-
HopSkipJump attack [4]	1000	14.56	0.062	6	-
	5000	4.01	0.031	17	-
	20000	1.85	0.012	42	-
GeoDA-fullspace	1000	11.76	0.053	6	-
	5000	3.35	0.022	10	-
	20000	1.06	0.009	14	-
GeoDA-subspace	1000	8.16	0.022	6	-
	5000	2.51	0.008	10	-
	20000	1.01	0.003	14	-
DeepFool (white-box) [24]	-	0.026	-	2	20
C&W (white-box) [2]	-	0.034	-	10000	10000

Table 2: The performance comparison of GeoDA with BA and HSJA for median ℓ_2 and ℓ_∞ on ImageNet dataset.

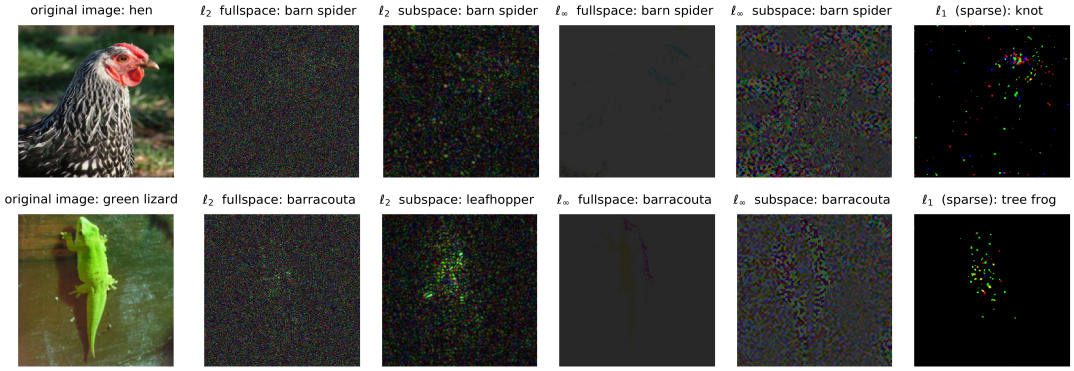
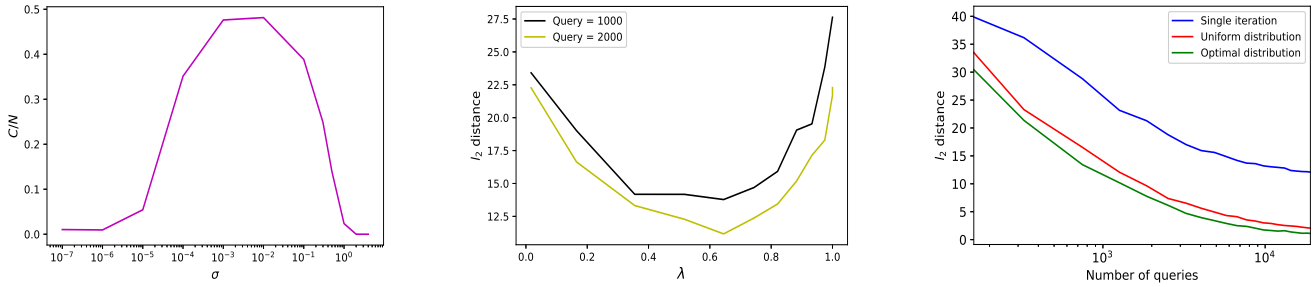


Figure 3: Original images and adversarial perturbations generated by GeoDA for ℓ_2 fullspace, ℓ_2 subspace, ℓ_∞ fullspace, ℓ_∞ subspace, and ℓ_1 sparse with $N = 10000$ queries. (Perturbations are magnified $\sim 10\times$ for better visibility.)

value for ζ, the higher the fooling rate and the sparsity and vice versa. In other words, choosing small values for ζ produces sparser adversarial examples; however, it decreases the chance that it is an adversarial example for the actual boundary. In Fig. 2c, we depicted the trade-off between fooling rate and sparsity by increasing the value for ζ for different query budgets. The larger the number of queries, the closer the initial point to the original image, and also the better our algorithm performs in generating sparse adversarial examples. In Table 1, the sparse GeoDA is compared with the white-box attack SparseFool. We show that with a limited number of queries, GeoDA can generate sparse perturbations with acceptable fooling rate with sparsity of about 3 percent with respect to the white-box attack SparseFool. The adversarial perturbations generated by GeoDA for ℓ_p norms are shown in Fig. 3 and the effect of different norms can be observed.

Incorporating prior information. Here, we evaluate the methods proposed in Section 4 to incorporate prior information in order to improve the estimation of the normal vector to the decision boundary. As sub-space priors, we deploy the DCT basis functions in which m low frequency subspace directions are chosen [22]. As shown in Fig. 5, biasing the search space to the DCT sub-space can reduce the ℓ_2 norm of the perturbations by approximately 27% compared to the full-space case. For transferrability, we obtain the normal vector of the given image using the white box attack DeepFool [24] on a ResNet-34 classifier. We bias the



(a) (b) (c)

Figure 4: (a) The effect of the variance σ on the ratio of correctly classified queries C to the total number of queries N at boundary point $\boldsymbol{x}_B$. (b) Effect of λ on the performance of the algorithm. (c) Comparison of two extreme cases of query distributions, i.e., single iteration ($\lambda \to 0$) and uniform distribution ($\lambda = 1$) with optimal distribution ($\lambda = 0.6$).

search space for normal vector estimation as described in Section 4. As it can be seen in Fig. 5, prior information can improve the normal vector estimation significantly.

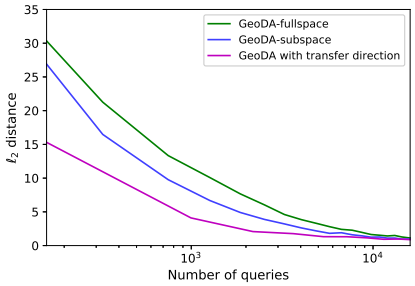


Figure 5: Effect of prior information, i.e., DCT sub-space and transferability on the performance of ℓ_2 perturbation.

6.2. Effect of hyper-parameters on the performance

In practice, we need to choose σ such that the locally flat assumption of the boundary is preserved. Upon generating the queries at boundary point $\boldsymbol{x}_B$ to estimate the direction of the normal vector as in (7), the value for σ is chosen in such a way that the number of correctly classified images and adversarial images on the boundary are almost the same. In Fig. 4a, the effect of variance σ of added Gaussian perturbation on the number of correctly classified queries on the boundary point is illustrated. We obtained a random point $\boldsymbol{x}_B$ on the decision boundary of the image classifier and query the image classifier 1000 times. As it can be seen, the variance σ is too small, none of the queries is correctly classified as the point $\boldsymbol{x}_B$ is not exactly on the boundary. On the other hand if the variance is too high, all the images are classified as adversarial since they are highly perturbed.

In order to obtain the optimal query distribution for a given limited budget N, the values for λ and T should be given. Having fixed λ, if T is large, the number of queries allocated to the first iteration may be too small. To address this, we consider a fixed number of queries for the first iteration as $N_1^* = 70$. Thus, having fixed λ, a reasonable choice for T can be obtained by solving (19) for T. Based on (19), if $\lambda \to 0$, all the queries are allocated to the last iteration and when $\lambda = 1$, the query distribution is uniform. A value between these two extremes is desirable for our algorithm. To obtain this value, we run our algorithm for different λ for only 10 images different from $\mathcal{X}$. Instead of throwing out the gradient obtained from the previous iterations, we can take advantage of them in next iterations as well. As it can be seen in Fig. 4b, the algorithm has its worst performance when λ is close to the two extreme cases: single iteration ($\lambda \to 0$) and uniform distribution ($\lambda = 1$). We thus choose the value $\lambda = 0.6$ for our experiments. Finally, in Fig. 4c, the comparison between three different query distributions is shown. The optimal query distribution achieves the best performance while the single iteration preforms worst. Actually, this fact is reflected in our proposed bound in (17) as even with infinite number of queries it can not do better than $\lambda(r_0 - r)$. Indeed the effect of curvature can be addressed only by increasing the number of iterations.

7. Conclusion

In this work, we propose a new geometric framework for designing query-efficient decision-based black-box attacks, in which the attacker only has access to the top-1 label of the classifier. Our method relies on the key observation that the curvature of the decision boundary of deep networks is small in the vicinity of data samples. This permits to estimate the normals to the decision boundary with a small number of queries to the classifier, hence to eventually design query-efficient ℓ_p-norm attacks. In the particular case of ℓ_2-norm attacks, we show theoretically that our algorithm converges to the minimal adversarial perturbations, and that the number of queries at each step of the iterative search can be optimized mathematically. We finally study GeoDA through extensive experiments that confirm its superior performance compared to state-of-the-art black-box attacks.

Acknowledgements

Supported in part by the US National Science Foundation under grants ECCS-1444009 and CNS-1824518. S. M. is supported by a Google Postdoctoral Fellowship.

References

[1] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint arXiv:1712.04248*, 2017. 1, 2, 6, 7

[2] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57, 2017. 1, 2, 7

[3] Jianbo Chen and Michael I Jordan. Boundary attack++: Query-efficient decision-based adversarial attack. *arXiv preprint arXiv:1904.02144*, 2019. 1, 2

[4] Jianbo Chen, Michael I Jordan, and Martin J Wainwright. Hopskipjumpattack: A query-efficient decision-based attack. *arXiv preprint arXiv:1904.02144*, 2019. 2, 6, 7

[5] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 15–26. ACM, 2017. 1, 2

[6] Steven Chen, Nicholas Carlini, and David Wagner. Stateful detection of black-box adversarial attacks. *arXiv preprint arXiv:1907.05587*, 2019. 5

[7] Minhao Cheng, Thong Le, Pin-Yu Chen, Jinfeng Yi, Huan Zhang, and Cho-Jui Hsieh. Query-efficient hard-label black-box attack: An optimization-based approach. *arXiv preprint arXiv:1807.04457*, 2018. 2, 6

[8] Shuyu Cheng, Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Improving black-box adversarial attacks with a transfer-based prior. *arXiv preprint arXiv:1906.06919*, 2019. 4

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255, 2009. 6

[10] Aditya Devarakonda, Maxim Naumov, and Michael Garland. Adabatch: adaptive batch sizes for training deep neural networks. *arXiv preprint arXiv:1712.02029*, 2017. 5

[11] Alhussein Fawzi, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. Robustness of classifiers: from adversarial to random noise. In *Advances in Neural Information Processing Systems*, pages 1632–1640, 2016. 2, 5

[12] Alhussein Fawzi, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. The robustness of deep networks: A geometrical perspective. *IEEE Signal Processing Magazine*, 34(6):50–62, 2017. 2

[13] Alhussein Fawzi, Seyed-Mohsen Moosavi-Dezfooli, Pascal Frossard, and Stefano Soatto. Empirical study of the topology and geometry of deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3762–3770, 2018. 2

[14] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 1, 2

[15] Chuan Guo, Jacob R Gardner, Yurong You, Andrew Gordon Wilson, and Kilian Q Weinberger. Simple black-box adversarial attacks. *arXiv preprint arXiv:1905.07121*, 2019. 3

[16] David Lee Hanson and Farroll Tim Wright. A bound on tail probabilities for quadratic forms in independent random variables. *The Annals of Mathematical Statistics*, 42(3):1079–1083, 1971. 4

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[18] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. *arXiv preprint arXiv:1804.08598*, 2018. 1, 2

[19] Andrew Ilyas, Logan Engstrom, and Aleksander Madry. Prior convictions: Black-box adversarial attacks with bandits and priors. *arXiv preprint arXiv:1807.07978*, 2018. 2, 3

[20] Yujia Liu, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. A geometry-inspired decision-based attack. *arXiv preprint arXiv:1903.10826*, 2019. 1, 2, 4, 6

[21] Gábor Lugosi, Shahar Mendelson, et al. Sub-gaussian estimators of the mean of a random vector. *The Annals of Statistics*, 47(2):783–794, 2019. 4

[22] Seyed Mohsen Moosavi Dezfooli. Geometry of adversarial robustness of deep networks: methods and applications. Technical report, EPFL, 2019. 7

[23] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1765–1773, 2017. 4

[24] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016. 1, 2, 7

[25] Nina Narodytska and Shiva Prasad Kasiviswanathan. Simple black-box adversarial perturbations for deep networks. *arXiv preprint arXiv:1612.06299*, 2016. 1, 2

[26] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 1

[27] GM Tallis. Plane truncation in normal populations. *Journal of the Royal Statistical Society: Series B (Methodological)*, 27(2):301–307, 1965. 3

[28] Florian Tramèr, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. The space of transferable adversarial examples. *arXiv preprint arXiv:1704.03453*, 2017. 4

[29] Chun-Chen Tu, Paishun Ting, Pin-Yu Chen, Sijia Liu, Huan Zhang, Jinfeng Yi, Cho-Jui Hsieh, and Shin-Ming Cheng. Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. *arXiv preprint arXiv:1805.11770*, 2018. 2

[30] Chun-Chen Tu, Paishun Ting, Pin-Yu Chen, Sijia Liu, Huan Zhang, Jinfeng Yi, Cho-Jui Hsieh, and Shin-Ming Cheng. Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 742–749, 2019. 2

[31] Pu Zhao, Sijia Liu, Pin-Yu Chen, Nghia Hoang, Kaidi Xu, Bhavya Kailkhura, and Xue Lin. On the design of black-box adversarial examples by leveraging gradient-free optimization and operator splitting method. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 121–130, 2019. 2