Universalization of Any Adversarial Attack using Very Few Test Examples

Sandesh Kamath
sandesh.kamath@cse.iith.ac.in
Indian Institute of Technology
Hyderabad, Telangana, India

Amit Deshpande
amitdesh@microsoft.com
Microsoft Research
Bengaluru, India

K V Subrahmanyam
kv@cmi.ac.in
Chennai Mathematical Institute
Chennai, Tamil Nadu, India

Vineeth N Balasubramanian
vineethnb@iith.ac.in
Indian Institute of Technology
Hyderabad, Telangana, India

ABSTRACT

Deep learning models are known to be vulnerable not only to input-dependent adversarial attacks but also to input-agnostic or universal adversarial attacks. Dezfooli et al. [8, 9] construct universal adversarial attack on a given model by looking at a large number of training data points and the geometry of the decision boundary near them. Subsequent work [5] constructs universal attack by looking only at test examples and intermediate layers of the given model. In this paper, we propose a simple universalization technique to take any input-dependent adversarial attack and construct a universal attack by only looking at very few adversarial test examples. We do not require details of the given model and have negligible computational overhead for universalization. We theoretically justify our universalization technique by a spectral property common to many input-dependent adversarial perturbations, e.g., gradients, Fast Gradient Sign Method (FGSM) and DeepFool. Using matrix concentration inequalities and spectral perturbation bounds, we show that the top singular vector of input-dependent adversarial directions on a small test sample gives an effective and simple universal adversarial attack. For standard models on CIFAR10 and ImageNet, our simple universalization of Gradient, FGSM, and DeepFool perturbations using a test sample of 64 images gives fooling rates comparable to state-of-the-art universal attacks [5, 9] for reasonable norms of perturbation.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; *Neural networks*; • **Security and privacy** → *Intrusion/anomaly detection and malware mitigation*;

KEYWORDS

adversarial, universal, neural networks

1 INTRODUCTION

Neural network models achieve high accuracy on several image classification tasks but are also known to be vulnerable to adversarial attacks. Szegedy et al. [13] showed that tiny pixel-wise changes in images, although imperceptible to the human eye, make highly accurate neural network models grossly misclassify. For a given classifier f, an adversarial attack $\mathcal{A}$ perturbs each input x by a carefully chosen small perturbation $\mathcal{A}(x)$ that changes the predicted label as $f(x + \mathcal{A}(x)) \neq f(x)$, for most inputs. Most adversarial attacks are *input-dependent*, i.e., $\mathcal{A}(x)$ depends on x. If the underlying model parameters θ for the classifier f are trained to minimize certain loss function $L(\theta, x, y)$ on data point x with label y, then perturbing along the gradient $\nabla_x L(\theta, x, y)$ is a natural adversary for maximizing loss, and hopefully, changing the predicted label. If an adversarial attack $\mathcal{A}$ changes each pixel value by at most $\pm\epsilon$, then its ℓ_∞-norm is bounded as $\|\mathcal{A}(x)\|_\infty \leq \epsilon$, for all x. Szegedy et al. [13] showed that it is possible to find such a perturbation using box-constrained L-BFGS. Goodfellow et al [3] proposed the Fast Gradient Sign Method (FGSM) using $\mathcal{A}(x) = \epsilon \operatorname{sign}(\nabla_x L(\theta, x, y))$ as a faster approach to find such an adversarial perturbation. Subsequent work on FGSM includes an iterative variant by Kurakin et al [6] and another version called Projected Gradient Descent (PGD) by Madry et al [7], both of which constructed adversarial perturbations with bounded ℓ_∞-norm. On the other hand, Deep-Fool by Moosavi-Dezfooli et al. [10] computed a minimal ℓ_2-norm adversarial perturbation iteratively. In each iteration, it used a polyhedron $\mathcal{P}_t$ to approximate a region around the current iterate $x^{(t)}$, where the classifier output is the same as $f(x^{(t)})$. The next iterate $x^{(t+1)}$ was the projection of $x^{(t)}$ on to the nearest face of $\mathcal{P}_t$. The algorithm was terminated when $f(x^{(t)}) \neq f(x)$, so the perturbation produced by DeepFool on input x is $x^{(t)} - x$. All *input-dependent* adversarial attacks mentioned above can be executed at test time with access only to the given model but not its training data.
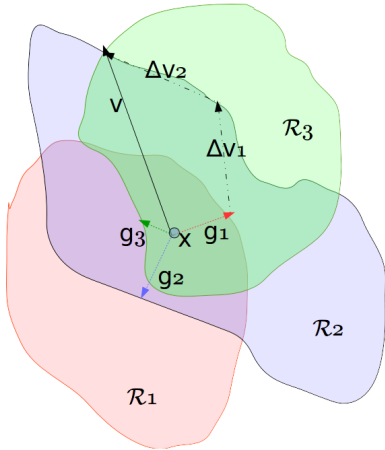
Figure 1: Illustration of the universal adversarial attack problem.

Universal adversarial perturbations are *input-agnostic*, i.e., a given model gets fooled into misclassification by the same perturbation on a large fraction of inputs. Moosavi-Dezfooli et al. [9] constructed a universal adversarial attack by clever, iterative calls to their input-dependent DeepFool attack. They theoretically justified the phenomenon of universal adversarial perturbations using certain geometric assumptions about the decision boundary [8]. Given a data distribution $\mathcal{D}$, a universal adversarial perturbation is a vector v of *small* ℓ_2-norm such that $f(x + v) \neq f(x)$, with high probability (called the *fooling rate*), for test input x sampled from the distribution $\mathcal{D}$. For a given bound ϵ on the ℓ_2-norm of universal adversarial perturbation and a given desired fooling rate, Moosavi-Dezfooli et al. [9] considered a sample S of training data, initialized $v = \bar{0}$, and proceeded iteratively as follows: if the fraction of $x \in S$ for which $f(x + v) \neq f(x)$ is less than the desired fooling rate, then they pick an x such that $f(x+v) = f(x)$, and find a minimal ℓ_2-norm perturbation Δv_x such that $f(x + v + \Delta v_x) \neq f(x + v)$ using Deep-Fool. Then they update v to $v + \Delta v_x$ and scale it down, if required, to have its ℓ_2-norm bounded by ϵ. Figure 1 gives an illustration of their approach on points x_1, x_2, x_3 belonging to distinct classes, shown in three colors. For visualization purposes, the regions containing these points are shown overlapped at points x_1, x_2, x_3, which is the point labeled x in the figure. Let g_1, g_2, g_3 be the minimal ℓ_2-norm perturbations such that $f(x_i + g_i) \neq f(x_i)$. Moosavi-Dezfooli et al. [9] iteratively identified adversarial perturbations Δv_1 and Δv_2 such that $f(x_2 + g_1 + \Delta v_1) \neq f(x_2)$ and $f(x_3 + g_1 + \Delta v_1 + \Delta v_2) \neq f(x_3)$. In Figure 1, $v = g_1 + \Delta v_1 + \Delta v_2$ achieves $f(x_i + v) \neq f(x_i)$, for $i = 1, 2, 3$, simultaneously. Moosavi-Dezfooli et al. [9] showed that the universal adversarial perturbation constructed as above from a large sample of training data gave a good fooling rate even on test data. Note that the above construction of universal adversarial attack requires access to training data and several iterations of the DeepFool attack.

The above discussion raises some natural, important questions: (a) Is there a simpler construction to *universalize* any given input-dependent adversarial attack? (b) Can a universal attack be constructed efficiently using access to the model and very few test inputs, with no access to the training data at all (or the test data in entirety)? We answer both of these questions affirmatively.

Our key results are summarized as follows:

• Our first observation is that many known input-dependent adversarial attack directions have only a small number of dominant principal components on the entire data. We firstly show this for attacks based on the gradient of the loss function, the FGSM attack, and DeepFool, on different architectures and datasets.

• Consider a matrix whose each row corresponds to input-dependent adversarial direction for a test data point. Our second observation is that a small perturbation along the top principal component of this matrix is an effective universal adversarial attack. This simple approach using Singular Value Decomposition (SVD), our SVD-Universal algorithm combined with Gradient, FGSM and DeepFool directions gives us SVD-Gradient, SVD-FGSM and SVD-DeepFool universal adversarial attacks, respectively.

• Our third observation is that the top principal component can be well-approximated from a very small sample of the test data (following [5]), and SVD-Universal approximated from even a small sample gives a fooling rate comparable to Moosavi-Dezfooli et al. [9]. Importantly, this approach can be used with any attack, as we show with three different methods in this work.

• We give a theoretical justification of this phenomenon using matrix concentration inequalities and spectral perturbation bounds. This observation holds across multiple input-dependent adversarial attack directions given by Gradient, FGSM and DeepFool.

2 RELATED WORK

The previous works closest to ours are the universal adversarial attacks by Moosavi-Dezfooli et al. [9] and Khrulkov and Oseledets [5]. Our approach to construct a universal adversarial attack is to take an input-dependent adversarial attack on a given model, and then find a single direction via SVD that is simultaneously well-aligned with the different input-dependent attack directions for most test data points. This is different from the approach of Moosavi-Dezfooli et al. [9] explained in Figure 1. When a training data point is not fooled by a smaller perturbation in previous iterations, Moosavi-Dezfooli et al. [9] apply DeepFool to such already-perturbed but unfooled data points. In contrast, our universalization uses input-dependent attack directions only on a small sample of data points, and even the simple universalization of gradient directions (instead of DeepFool) already gives a comparable fooling rate to the universal attack of Moosavi-Dezfooli et al. [9] in our experiments.

Khrulkov and Oseledets [5] propose a state-of-the-art universal adversarial attack that requires expensive computation and access to the hidden layers of the given neural network model. They consider the function $f_i(x)$ computed by the i-th hidden layer on input x, and its Jacobian $J_i(x) = \partial f_i / \partial x \big|_x$. Using $\|f_i(x + v) - f_i(x)\|_q \approx \|J_i(x)v\|_q$, they solve a (p, q)-SVD problem to maximize $\sum_{x \in \mathcal{X}} \|J_i(x)v\|_q^q$ subject to $\|\|_p = 1$ over the entire data $\mathcal{X}$. They optimize for the choice of layer i and (p, q) in the (p, q)-SVD for the i-th hidden layer. They *hypothesize* that this objective can be

empirically approximated by a sample S of size m from test data (see Eqn.(8) in [5]). With extensive experiments on ILSVRC 2012 validation data for VGG-16, VGG-19 and ResNet50 models, they empirically find the best layer i to attack and the empirically best choice of (p, q) (e.g., $q = 10$ for $p = \infty$). We do not solve the general (p, q)-SVD, which is known to be NP-hard for most choices of (p, q) [1, 2]. Our SVD-Universal algorithm uses the regular SVD $(p = q = 2)$, which can be solved provably and efficiently. Our method does not require access to hidden layers, and it universalizes several known input-dependent adversarial perturbations. We *prove* that our objective can be well-approximated from only a small sample of test data (Theorem 2), following Khrulkov and Oseledets [5], who however only *hypothesize* this for their objective (see Eqn.(8) in [5]).

Recent work has also considered model-agnostic and data-agnostic adversarial perturbations. Tramer et al. [14] study model-agnostic perturbations in the direction of the difference between the intra-class means, and come up with adversarial attacks that transfer across different models. Mopuri et al. [11] propose a data-agnostic adversarial attack that depends only on the model architecture. Given a trained neural network with k hidden (convolution) layers, they start with a random image v and minimize $\prod_{i=1}^{k} \ell_i(v)$ subject to $\|v\|_\infty \le \epsilon$, where $\ell_i(v)$ is the mean activation of the i-th hidden layer for input v. The authors show that the optimal perturbation v for this objective exhibits a data-agnostic adversarial attack. In contrast to these methods, we present a simple yet effective method based on the principal component of a few attack directions, as described further below.

3 SVD-UNIVERSAL: A SIMPLE METHOD TO UNIVERSALIZE AN ADVERSARIAL ATTACK

We begin by defining the notation and the evaluation metric *fooling rate* formally. Let $\mathcal{D}$ denote the data distribution on image-label pairs (x, y), with images as a d-dimensional vectors in some $\mathcal{X} \subseteq \mathbb{R}^d$ and labels in $[k] = \{1, 2, \ldots, k\}$ for k-class classification, e.g., CIFAR-10 data has images with 32×32 pixels that are essentially 1024-dimensional vectors of pixel values, each in $[0, 1]$, along with their respective labels for 10-class classification. Let (X, Y) be a random data point from $\mathcal{D}$ and let $f : \mathcal{X} \to [k]$ be a k-class classifier. We use θ to denote the model parameters for classifier f, and let $L(\theta, x, y)$ denote the loss function it minimizes on the training data. The accuracy of classifier f is given by $\Pr_{(X,Y)}(f(X) = Y)$. A classifier f is said to be fooled on input x by adversarial perturbation $\mathcal{A}(x)$ if $f(x + \mathcal{A}(x)) \ne f(x)$. The fooling rate of the adversary $\mathcal{A}$ is defined as $\Pr_{(X,Y)}(f(X + \mathcal{A}(X)) \ne f(X))$.

Our approach to construct a universal adversarial attack is to take an input-dependent adversarial attack on a given model, and then find a single direction via SVD that is simultaneously well-aligned with the different input-dependent attack directions for most test data points. We apply this approach to a very small sample (less than 0.2%) of test data, and use the top singular vector as a universal adversarial direction. Our algorithm, SVD-Universal, is presented in Algorithm 1. We prove that if the input-dependent attack directions satisfy a certain spectral property, then our approach can provably result in a good fooling rate (Theorem 1), and we prove that a

---

Algorithm 1: SVD-Universal Algorithm

Data: A neural network N, an input-dependent adversarial attack $\mathcal{A}$, and n test samples.

Result: A universal attack direction for neural network N

1 For test samples $x_1, x_2, \ldots, x_n$, obtain input-dependent perturbation vectors $a_1 = \mathcal{A}(x_1), a_2 = \mathcal{A}(x_2), \ldots, a_n = \mathcal{A}(x_n)$ for the neural network N.

2 Normalize a_i's to get the attack directions or unit vectors $u_i = a_i / \|a_i\|_2$, for $i = 1$ to n.

3 Form a matrix M whose rows are $u_1, u_2, \ldots, u_n$.

4 Compute Singular Value Decomposition (SVD) of M as $M = USV^T$, with $V = [v_1|v_2|\ldots|v_n]$.

5 Return the top right singular vector v_1 as the universal attack vector.

---

small sample size suffices, independent of the data dimensionality (Theorem 2).

Our SVD-Universal algorithm is flexible enough to universalize many popular input-dependent adversarial attacks. We apply it in three different ways to construct input-dependent perturbations: (a) Gradient attack that perturbs an input x in the direction $\nabla_x L(\theta, x, y)$, (b) FGSM attack [3] that perturbs x in the direction $\text{sign}(\nabla_x L(\theta, x, y))$, (c) DeepFool attack [10] which is an iterative algorithm explained in Section 1. For the above three input-dependent attacks, we call the universal adversarial attack produced by SVD-Universal as SVD-Gradient, SVD-FGSM, and SVD-DeepFool, respectively.

As shown in Algorithm 1, SVD-Universal samples a set of n images from the test (or validation) set. We use the terms batch size and sample size interchangeably. For each of the sampled points, we compute an input-dependent attack direction. We stack these attack directions as rows of a matrix. The top right singular vector of this matrix of attack directions is the universal adversarial direction that SVD-Universal outputs.

4 THEORETICAL ANALYSIS OF SVD-UNIVERSAL

In this section, we provide a theoretical justification for the existence of universal adversarial perturbations. Let (X, Y) denote a random sample from $\mathcal{D}$. Let $f : \mathbb{R}^d \to [k]$ be a given classifier, and for any $x \in \mathbb{R}^d$, let $\mathcal{A}(x)$ be the adversarial perturbation given by a fixed attack $\mathcal{A}$, say *FGSM, DeepFool*.

Define $A = \{x : f(x + \mathcal{A}(x)) \ne f(x)\}$. For any $x \in A$, assume that $x + \mathcal{A}(x)$ lies on the decision boundary, and let the hyperplane $H_x = \{x + z \in \mathbb{R}^d : \langle z, \mathcal{A}(x) \rangle = \|\mathcal{A}(x)\|_2^2\}$ be a local, linear approximation to the decision boundary at $x + \mathcal{A}(x)$. This holds for adversarial attacks such as DeepFool by Moosavi-Dezfooli et al. [10] which try to find an adversarial perturbation $\mathcal{A}(x)$ such that $x + \mathcal{A}(x)$ is the nearest point to x on the decision boundary. Now consider the halfspace $S_x = \{x + z \in \mathbb{R}^d : \langle z, \mathcal{A}(x) \rangle \ge \|\mathcal{A}(x)\|_2^2\}$. Note that $x \notin S_x$ and $x + \mathcal{A}(x) \in S_x$. *For simplicity of analysis, we assume that $f(x + z) \ne f(x)$, for all $x \in A$ and $x + z \in S_x$.* This is a reasonable assumption in a small neighborhood of x. In fact, this hypothesis is implied by the positive curvature of the

decision boundary assumed in the analysis of Moosavi-Dezfooli et al. [8]. Moosavi-Dezfooli et al. [8] empirically verify the validity of this hypothesis. In other words, we assume that if an adversarial perturbation $\mathcal{A}(x)$ fools the model at x, then any perturbation z having a sufficient projection along $\mathcal{A}(x)$, also fools the model at x. This is a reasonable assumption in a small neighborhood of x.

THEOREM 1. *Given any joint data distribution $\mathcal{D}$ on features or inputs in $\mathbb{R}^d$ and true labels in $[k]$, let (X, Y) denote a random sample from $\mathcal{D}$. For any $x \in \mathbb{R}^d$, let $\mathcal{A}(x)$ be its adversarial perturbation by a fixed input-dependent adversarial attack $\mathcal{A}$. Let*

$$M = \mathbb{E}\left[\frac{\mathcal{A}(X)}{\|\mathcal{A}(X)\|_2}\frac{\mathcal{A}(X)^T}{\|\mathcal{A}(X)\|_2}\right] \in \mathbb{R}^{d \times d},$$

and $0 \le \lambda \le 1$ be the top eigenvalue of M and $v \in \mathbb{R}^d$ be the normalized unit eigenvector. Then, for any $0 < \delta < \sqrt{\lambda}$, under the assumption that $f(x+z) \ne f(x)$, for all $x \in A$ and $x+z \in S_x$, we have

$$\Pr\left(f(X + u) \ne f(X)\right) \ge \Pr\left(f(X + \mathcal{A}(X)) \ne f(X)\right) - \frac{1-\lambda}{1-\delta^2},$$

where $u = \pm(\epsilon/\delta)v$, where $\epsilon = \max_x \|\mathcal{A}(x)\|_2$.

PROOF. Let $\mu(x)$ denote the induced probability density on features or inputs by the distribution $\mathcal{D}$. Define $A = \{x : f(x + \mathcal{A}(x)) \ne f(x)\}$ and $G = \{x : |\langle\mathcal{A}(x), v\rangle| \ge \delta \|\mathcal{A}(x)\|_2\}$. Since λ is the top eigenvalue of M with v as its corresponding (unit) eigenvector,

$$\lambda = \mathbb{E}\left[\left\langle\frac{\mathcal{A}(X)}{\|\mathcal{A}(X)\|_2}, v\right\rangle^2\right]$$

$$= \int_{x \in G}\left\langle\frac{\mathcal{A}(x)}{\|\mathcal{A}(x)\|_2}, v\right\rangle^2\mu(x)dx + \int_{x \notin G}\left\langle\frac{\mathcal{A}(x)}{\|\mathcal{A}(x)\|_2}, v\right\rangle^2\mu(x)dx$$

$$\le \int_{x \in G}\mu(x)dx + \delta^2\int_{x \notin G}\mu(x)dx \quad \text{because } \|v\|_2 = 1$$

$$= \Pr(G) + \delta^2(1 - \Pr(G))$$

$$= (1 - \delta^2)\Pr(G) + \delta^2.$$

Thus, $\Pr(G) \ge (\lambda - \delta^2)/(1 - \delta^2)$, and equivalently, $\Pr(G^c) = 1 - \Pr(G) \le (1-\lambda)/(1-\delta^2)$. Now for any $x \in G$, we have $|\langle\mathcal{A}(x), v\rangle| \ge \delta \|\mathcal{A}(x)\|_2$. Letting $\epsilon = \max_x \|\mathcal{A}(x)\|_2$, we get $|\langle\mathcal{A}(x), (\epsilon/\delta)v\rangle| \ge \|\mathcal{A}(x)\|_2^2$. Thus, $x \pm (\epsilon/\delta)v \in S_x$, where $S_x = \{x + z \in \mathbb{R}^d : \langle z, \mathcal{A}(x)\rangle \ge \|\mathcal{A}(x)\|_2^2\}$, and therefore, by our assumption stated before Theorem 1, we have $f(x) \ne f(x + u)$, where $u = \pm(\epsilon/\delta)v$. Putting all of this together, $\Pr(f(X + u) \ne f(X)) \ge \Pr(G \cap A) \ge \Pr(A) - \Pr(A \cap G^c) \ge \Pr(A) - \Pr(G^c)$, and therefore, $\Pr(f(X + u) \ne f(X)) \ge \Pr(f(X + \mathcal{A}(X)) \ne f(X)) - \frac{1-\lambda}{1-\delta^2}$. □

Theorem 1 shows that any norm-bounded, input-dependent, adversarial attack $\mathcal{A}$ can be converted into a universal attack u of comparable norm, without losing much in the fooling, if the top eigenvalue λ of M is close to 1. This universal attack direction lies in the one-dimensional span of the top eigenvector v of M. The proof of Theorem 1 can be easily generalized to the top SVD subspace of M and where the top few eigenvalues of M dominate its spectrum (note that $\text{tr}(M) = 1$).
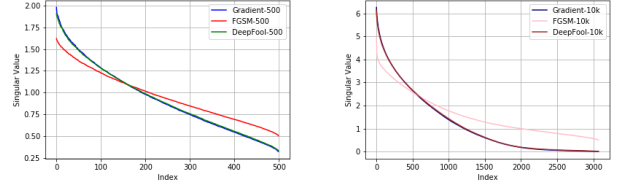

Figure 2: On CIFAR-10, ResNet18, Singular values of attack directions over a sample of (top) 500 and (bottom) 10,000 test points.

Singular value drop. We empirically verify our hypothesis about top eigenvalue (or the top few eigenvalues) dominating the spectrum of M in Theorem 1. Let $X_1, X_2, \ldots, X_m$ be m i.i.d. samples of X drawn from the distribution $\mathcal{D}$ and consider the unnormalized, empirical analog of M as follows:

$$\sum_{i=1}^m \frac{\mathcal{A}(X_i)}{\|\mathcal{A}(X_i)\|_2}\frac{\mathcal{A}(X_i)^T}{\|\mathcal{A}(X_i)\|_2}.$$

Figure (2) shows how the singular values drop for the three input dependent attacks, *Gradient*, *FGSM*, and *DeepFool* on CIFAR-10 trained on ResNet18 on batch sizes 500 and 10,000. These plots indicate that the drop in singular values is a shared phenomenon across different input-dependent attacks, and the trend is similar even when we look at a small number of input samples.

Our second contribution is finding a good approximation to the *universal* adversarial perturbation given by the top eigenvector v of M, using only a small sample $X_1, X_2, \ldots, X_m$ from $\mathcal{D}$. Theorem 2 shows that we can efficiently pick such a small sample whose size is independent of $\mathcal{D}$, depends linearly on the *intrinsic dimension* of M, and logarithmically on the feature dimension d.

THEOREM 2. *Given any joint data distribution $\mathcal{D}$ on features in $\mathbb{R}^d$ and true labels in $[k]$, let (X, Y) denote a random sample from $\mathcal{D}$. For any $x \in \mathbb{R}^d$, let $\mathcal{A}(x)$ denote the adversarial perturbation of x according a fixed input-dependent adversarial attack $\mathcal{A}$. Let*

$$M = \mathbb{E}\left[\frac{\mathcal{A}(X)}{\|\mathcal{A}(X)\|_2}\frac{\mathcal{A}(X)^T}{\|\mathcal{A}(X)\|_2}\right] \in \mathbb{R}^{d \times d}.$$

Let $0 \le \lambda = \|M\|_2 \le 1$ denote the top eigenvalue of M and let v denote its corresponding eigenvector (normalized to have unit ℓ_2 norm). Let $r = \text{tr}(M)/\|M\|_2$ be the intrinsic dimension of M. Let $X_1, X_2, \ldots, X_m$ be m i.i.d. samples of X drawn from the distribution $\mathcal{D}$, and let $\tilde{\lambda} = \|\tilde{M}\|_2$ be the top eigenvalue of the matrix $\tilde{M}$,

$$\tilde{M} = \frac{1}{m}\sum_{i=1}^m \frac{\mathcal{A}(X_i)}{\|\mathcal{A}(X_i)\|_2}\frac{\mathcal{A}(X_i)^T}{\|\mathcal{A}(X_i)\|_2},$$

and $\tilde{v}$ be the top eigenvector of $\tilde{M}$.

Also suppose that there is a gap of at least $\gamma\lambda$ between the top eigenvalue λ and the second eigenvalue of M. Then for any $0 \le \epsilon < \gamma$ and $m = O(\epsilon^{-2}r\log d)$, we get $\|v - \tilde{v}\|_2 \le \epsilon/\gamma$, with a constant probability. This probability can be boosted to $1 - \delta$ by having an additional $\log(1/\delta)$ in the $O(\cdot)$.

Proof. Take $m = O(\epsilon^{-2} r \log d)$. By the covariance estimation bound (see Vershynin [15, Theorem 5.6.1]) and Markov's inequality, we get that $\left\| M - \tilde{M} \right\|_2 \leq \epsilon \lambda$, with a constant probability. Applying Weyl's theorem on eigenvalue perturbation [15, Theorem 4.5.3], we get $\left| \lambda - \tilde{\lambda} \right| \leq \epsilon \lambda$. Moreover, if there is gap of at least $\gamma \lambda$ between the first and the second eigenvalue of M with $\gamma > \epsilon$, we can use the Davis-Kahan theorem [15, Theorem 4.5.5] to bound the difference between the eigenvectors as $\left\| v - \tilde{v} \right\|_2 \leq \epsilon / \gamma$, with a constant probability. Please see Appendix, and the book by [15] cited therein, for more details about the covariance estimation bound, Weyl's theorem, and Davis-Kahan theorem. □

The description of the results cited in the proof above are provided in the Appendix for clarity of reading. The theoretical bounds are weaker than our empirical observations on the number of test samples needed for the attack. We wish to highlight again that the bound in Theorem 2 is independent of the support of underlying data distribution $\mathcal{D}$ and depends logarithmically on the feature dimension d. There is more room to tighten our analysis using more properties of the data distribution and the spectral properties of M.

5 EXPERIMENTS AND RESULTS

Datasets. CIFAR-10 dataset consists of $60,000$ images of 32×32 size, divided into 10 classes: $40,000$ used for training, $10,000$ for validation and $10,000$ for testing. ImageNet refers to the ILSRVC 2012 dataset [12] which consists of images of 224×224 size, divided into 1000 classes. All experiments performed on neural network-based models were done using the validation set of ImageNet and test set of CIFAR-10 datasets.

Model Architectures. For the ImageNet based experiments, we use pre-trained networks of VGG16, VGG19 and ResNet50 architectures[1]. For the CIFAR-10 experiments, we use the ResNet18 architecture as in He et al. [4]. All of these are popularly used models. We used off the shelf code available for these architectures. In these architectures pixel intensities of images are scaled down and images are normalized before they are deployed for use in classifiers. Our (unit) attack vectors are constructed using batch size of 64 (0.13%). We found that SVD-Universal attacks obtained using batch size of 64 perform as well as SVD-Universal attacks obtained using larger batch sizes of 128 and 1024 (see below). We compare the results of our attacks with M-DFFF, which denotes the universal perturbation vecror obtained using the method of Moosavi-Dezfooli et al [9]. For fair comparison, the same 64 samples used to construct our SVD-Universal vectors were used to obtain the universal perturbation vector, M-DFFF, of Moosavi-Dezfooli et al [9]. This vector was scaled down to get a unit vector w in ℓ_2 norm. (Higher is better for all the presented results on error rate or fooling rate.)

SVD-Universal on ImageNet. In Figure 3, each plot has the fooling rate on VGG16, VGG19 and ResNet50 attacked by the same universal method. This shows the effectiveness of the same attack method on different networks. In Figure 4, to compare SVD-Universal and M-DFFF, we plot the fooling rate of both together for each network. Importantly, Figure 9 shows the fooling rates

[1]https://pytorch.org/docs/stable/torchvision/models.html

obtained with SVD-Universal with batch size of 64, 128, 1024 on VGG16, VGG19 and ResNet50, respectively. We observed that SVD-Universal attacks obtained using batch size of 64 perform as well as attack vectors obtained using larger batch sizes of 128 and 1024. We compare our fooling rate results with that of Moosavi-Dezfooli et al [9] on the validation set of ImageNet in Table 1.
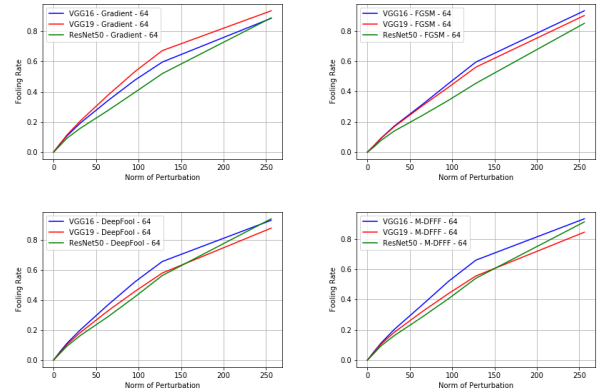

Figure 3: On ImageNet validation, VGG16 vs VGG19 vs ResNet50: fooling rate vs. norm of perturbation. Attacks constructed using 64 samples. (top left) *SVD-Gradient* **(top right)** *SVD-FGSM* **(bottom left)** *SVD-DeepFool* **and (bottom right) M-DFFF universal.**
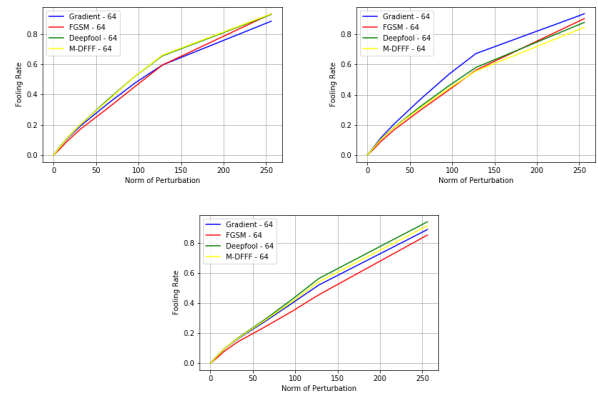

Figure 4: On ImageNet validation, *(top left):* **VGG16: fooling rate,** *(top right)* **VGG19: fooling rate,** *(bottom)* **ResNet50: fooling rate, vs. norm of perturbation along top singular vector of attack directions on 64 samples.**

SVD-Universal on CIFAR-10. We plot the error rates of SVD-Universal on CIFAR-10 in Figure 5 trained on ResNet18 with batch size of 100/500/10000. In Figure 6 we plot SVD-Universal and M-DFFF obtained with 100 samples for comparison. Similar to the observation made for ImageNet above, we see that the universal
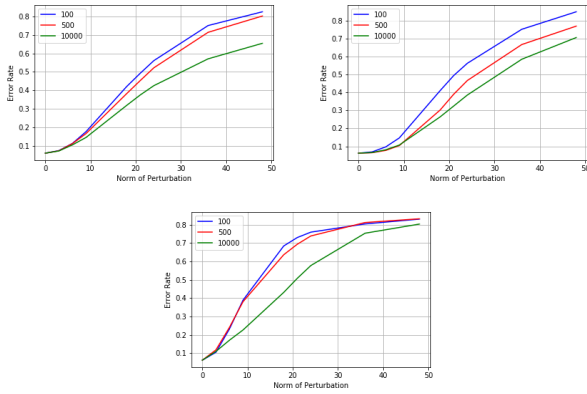
Figure 5: On CIFAR-10, ResNet18: error rate vs. norm of perturbation along top singular vector of attack directions on 100/500/10000 sample, *(top left)* Gradient *(top right)* FGSM *(bottom)* DeepFool
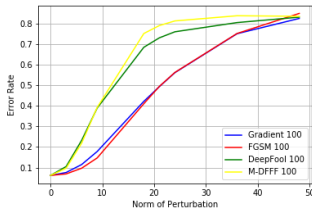

Figure 6: On CIFAR-10, ResNet18: error rate vs. norm of perturbation along top singular vector of attack directions on 100 samples.

attack with 100 samples performs comparable to the universal attack with larger batch size of 500 and 10000.

Our observations. (i) We observe a trend similar to what is reported by Khrulkov and Oseledets [5] - the fooling rate of SVD-Universal attacks is higher on VGG16 and VGG19 than on ResNet50. (ii) As noted earlier, we observe that SVD-Universal attacks obtained using batch size of 64 perform as well as attack vectors obtained using larger batch sizes of 128 and 1024. (iii) In Khrulkov and Oseledets [5, Figure 9], the authors report that the universal perturbation of [9] constructed from a batch size 64 and having ℓ_∞ norm 10 has a fooling rate of **0.14** on VGG19. A comparable perturbation in our model has ℓ_2 norm 4% of 450, and we get a fooling rate of **0.13** on VGG19. (iv) SVD-Gradient attack scaled to have norm 50 has a fooling rate of 0.32 on the validation set of ImageNet for VGG19. Note that the average ℓ_2 norm[2] of this validation set[3] is 450. (v) We visualize the perturbed images in Figure 7 when ϵ is 16 (3.5% of the average norm of input images) and when ϵ is 50 (11% of the average norm of input images). These perturbations are quasi-imperceptible [9].
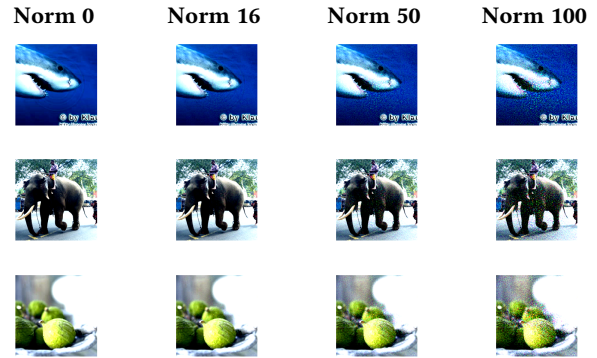

Figure 7: Sample images from ImageNet validation set perturbed with SVD-DeepFool of different l_2 norms.
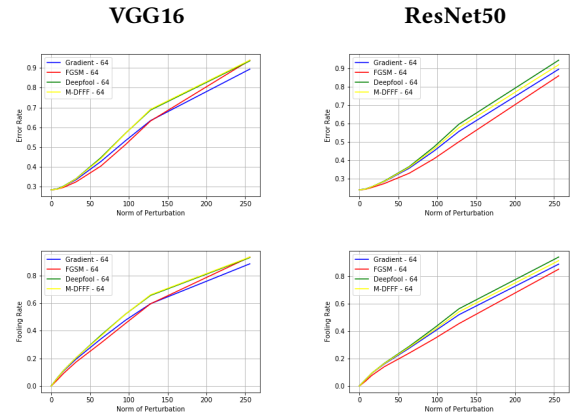

Figure 8: On ImageNet validation, (top left): VGG16: error rate (bottom left) VGG16: fooling rate (top right) ResNet50: error rate, (bottom right) ResNet50: fooling rate, vs. norm of perturbation along top singular vector of attack directions on 64 samples.

We note that Khrulkov and Oseledets [5] get a fooling rate of more than **0.4** using batch size of 64 and ℓ_∞ norm 10. Their universal attack is stronger than both our attack and that of Moosavi-Dezfooli et al. [9]. However, Khrulkov and Oseledets [5] do an extensive experimentation and determine which intermediate layer to attack and (p, q) are also optimized to maximize the fooling rate of their (p, q)-singular vector. (p, q)-SVD computation is expensive and is known to be a hard problem, [1, 2]. We do no such optimization and use $p = q = 2$, our emphasis being on the simplicity and universality of our SVD-Universal algorithm.

6 DISCUSSION

Connection between fooling rate and error rate. As stated earlier, let $\mathcal{D}$ be a distribution on pairs of images and labels, with images coming from a set $X \subseteq \mathbb{R}^d$. In the case of CIFAR-10 images, we can think of X to be the set of 32×32 CIFAR-10 images with pixel values from [0, 1]. So each image is a vector in a

Table 1: On ImageNet validation, VGG16 vs VGG19 vs ResNet50 vs M-DFFF: fooling rate vs. norm of perturbation. Attacks constructed using 64 samples.

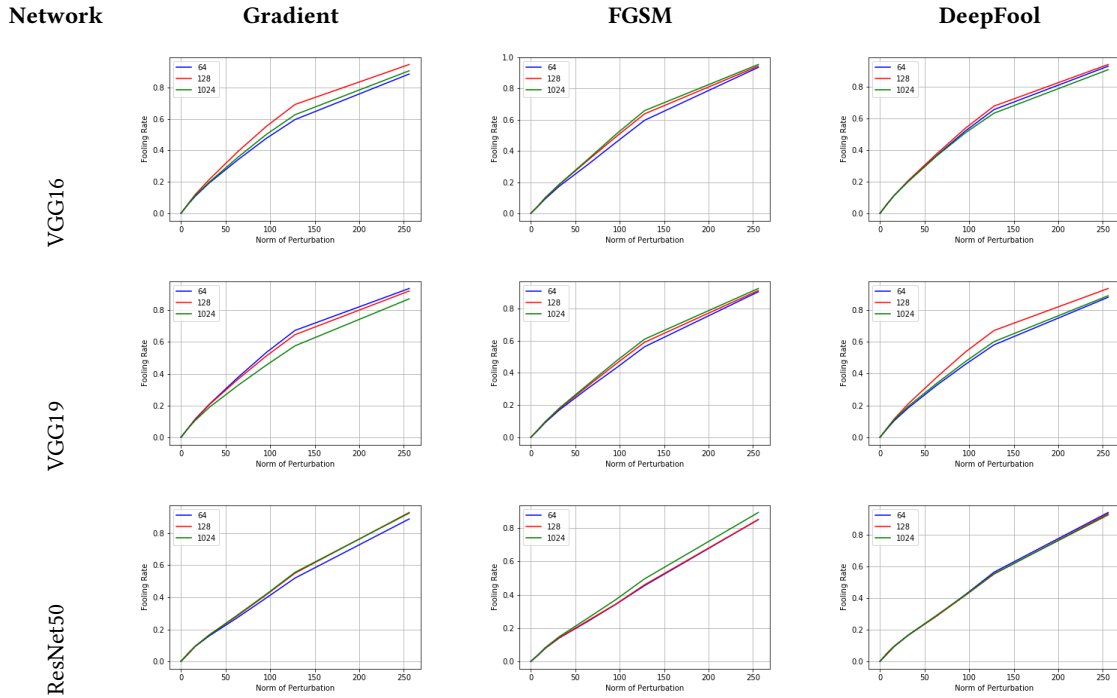| Network	Vector (using 64 samples)	Norm 18 (4%)	Norm 32 (7.1%)	Norm 64 (14.2%)
VGG16	SVD-Gradient	0.12	0.19	0.34
	SVD-FGSM	0.10	0.17	0.31
	SVD-DeepFool	**0.12**	**0.20**	**0.37**
	M-DFFF	0.11	0.20	0.36
VGG19	SVD-Gradient	0.13	0.21	0.38
	SVD-FGSM	0.10	0.17	0.30
	SVD-DeepFool	**0.11**	**0.19**	**0.33**
	M-DFFF	0.11	0.19	0.31
ResNet50	SVD-Gradient	0.10	0.16	0.28
	SVD-FGSM	0.09	0.14	0.24
	SVD-DeepFool	**0.10**	**0.17**	**0.29**
	M-DFFF	0.09	0.16	0.28

Figure 9: On ImageNet validation, fooling rate as per [9], on VGG16, VGG19 and ResNet50: fooling rate vs. norm of perturbation along top singular vector of attack directions on 64/128/1024 sample

space of dimension 1024. Let (X, Y) be a sample from $\mathcal{D}$ and let $f : \mathcal{X} \rightarrow [k]$, be a k-class classifier. The error rate of the classifier f is $\Pr_{(X,Y) \in \mathcal{D}}[f(X) \neq Y] = \beta$. An adversary $\mathcal{A}$ is a function $\mathcal{X} \rightarrow \mathbb{R}^d$. When $\mathcal{A}$ is a distribution over functions we get a randomized adversary. The norm of the perturbation applied to X is the norm of $\mathcal{A}(X)$ (we only consider ℓ_2 norm in this paper).

In Moosavi-Dezfooli et al. [8, 9] and Khrulkov and Oseledets [5], which we follow in this work for better comparison, the authors consider the fooling rate of an adversary. A classifier f is said to be fooled on input x by the perturbation $\mathcal{A}(x)$ if $f(x + \mathcal{A}(x)) \neq f(x)$.

The fooling rate of the adversary $\mathcal{A}$ is defined to be

$$\Pr_{(X,Y)}[f(X + \mathcal{A}(X)) \neq f(X)].$$

The adversarial error rate of $\mathcal{A}$ on the classifier f is defined to be $\Pr_{(X,Y) \in \mathcal{D}}[f(X + \mathcal{A}(X)) \neq Y]$. It is easy to see that

$$\Pr_{(X,Y)}[f(X + \mathcal{A}(X)) \neq f(X)] \geq \Pr_{(X,Y)}[f(X + \mathcal{A}(X)) \neq Y] - \beta.$$

So, if the natural accuracy of the classifier f is high, the fooling rate is close to the adversarial error rate. The error rate of the adversary with zero perturbation is the error rate of the trained network, whereas the fooling rate of the adversary with zero perturbation is necessarily zero. However, small fooling rate does not necessarily

imply small error rate, especially when the natural accuracy is not close to 100%. Note that existing models such as VGG16, VGG19, ResNet50 do not achieve natural accuracy greater than 0.8 on the ImageNet dataset. In Figure 8, we present the plots of SVD-Universal and M-DFFF on VGG16 and ResNet50 networks using fooling rate and error rate for comparison.

Visualizing SVD-Universal perturbations. We visualize the top singular vectors for the *Gradient*, *FGSM*, *DeepFool* directions from ImageNet and CIFAR-10 in Figure 10 and Figure 11, respectively. We observe that the *Gradient* and *DeepFool*-based singular directions are more concentrated in few regions, while the *FGSM* is more spread out. This observation could be useful in understanding the role of universal directions and adversarial robustness in general.
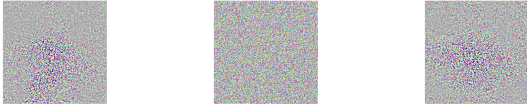

Figure 10: For ImageNet, Top SVD vector from (left) Gradient, (center) FGSM, (right) DeepFool on ResNet50.
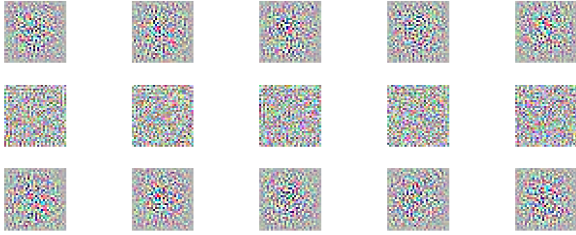

Figure 11: For CIFAR-10 (top) Top 5 SVD vectors from *Gradient*, (middle) Top 5 SVD vectors from *FGSM*, (bottom) Top 5 SVD vectors from DeepFool on ResNet18.

7 CONCLUSION

In this work, we show how to use a small sample of input-dependent adversarial attack directions on test inputs to find a single universal adversarial perturbation that fools state-of-the-art neural network models. Our main observation is a spectral property common to different attack directions such as *Gradients*, *FGSM*, *DeepFool*. We give a theoretical justification for how this spectral property helps in universalizing the adversarial attack directions by using the top singular vector. We justify theoretically and empirically that such a perturbation can be computed using only a small sample of test inputs.

ACKNOWLEDGMENTS

APPENDIX

The results below are used in the proof of Theorem 2, and are included herein for completeness. Theorem 5.6.1 (General covariance estimation) from [15] bounds the spectral norm of covariance matrix estimated from a small number of samples as follows.

THEOREM 3 ([15, THM 5.6.1]). *Let X be a random vector in $\mathbb{R}^d$, $d \geq 2$. Assume that for some $K \geq 1$, $\|X\|_2 \leq K \left(\mathbb{E}\left[\|X\|_2^2\right] \right)^{1/2}$, almost surely. Let $\Sigma = \mathbb{E}\left[XX^T\right]$ be the covariance matrix of X and $\Sigma_m = \frac{1}{m} \sum_{i=1}^m X_i X_i^T$ be the estimated covariance from m i.i.d. samples $X_1, X_2, \ldots, X_m$. Then for every positive integer m, we have*

$$\mathbb{E}\left[\|\Sigma_m - \Sigma\|_2\right] \leq C \left(\sqrt{\frac{K^2 d \log d}{m}} + \frac{K^2 d \log d}{m} \right) \|\Sigma\|_2 ,$$

for some positive constant C and $\|\Sigma\|_2$ being the spectral norm (or the top eigenvalue) of Σ.

Note that using $m = O(\epsilon^{-2} d \log d)$ we get $\mathbb{E}\left[\|\Sigma_m - \Sigma\|_2\right] \leq \epsilon \|\Sigma\|_2$. A tighter version of Theorem 5.6.1 appears as Remark 5.6.3, when the *intrinsic dimension* $r = \text{tr}(\Sigma) / \|\Sigma\|_2 \ll d$.

THEOREM 4 ([15, REMARK 5.6.3]). *Let X be a random vector in $\mathbb{R}^d$, and $d \geq 2$. Assume that for some $K \geq 1$, $\|X\|_2 \leq K \left(\mathbb{E}\left[\|X\|_2^2\right] \right)^{1/2}$, almost surely. Let $\Sigma = \mathbb{E}\left[XX^T\right]$ be the covariance matrix of X and $\Sigma_m = \frac{1}{m} \sum_{i=1}^m X_i X_i^T$ be the estimated covariance from m i.i.d. samples $X_1, X_2, \ldots, X_m$. Then for every positive integer m, we have*

$$\mathbb{E}\left[\|\Sigma_m - \Sigma\|_2\right] \leq C \left(\sqrt{\frac{K^2 r \log d}{m}} + \frac{K^2 r \log d}{m} \right) \|\Sigma\|_2 ,$$

for some positive constant C and $\|\Sigma\|_2$ being the operator norm (or the top eigenvalue) of Σ.

Note that using $m = O(\epsilon^{-2} r \log d)$ we get $\mathbb{E}\left[\|\Sigma_m - \Sigma\|_2\right] \leq \epsilon \|\Sigma\|_2$. Theorem 4.5.3 (Weyl's Inequality) from [15] upper bounds the difference between i-th eigenvalues of two symmetric matrices A and B using the spectral norm of $A - B$.

THEOREM 5 ([15, THM 4.5.3 (WEYL'S INEQUALITY)]). *For any two symmetric matrices A and B in $\mathbb{R}^{d \times d}$, $|\lambda_i(A) - \lambda_i(B)| \leq \|A - B\|_2$, where $\lambda_i(A)$ and $\lambda_i(B)$ are the i-th eigenvalues of A and B, respectively.*

In other words, the spectral norm of matrix perturbation bounds the stability of its spectrum. Here is a special case of Theorem 4.5.5 (Davis-Kahan Theorem) and its immediate corollary mentioned in [15].

THEOREM 6 ([15, THM 4.5.5 (DAVIS-KAHAN THEOREM)]). *Let A and B be symmetric matrices in $\mathbb{R}^{d \times d}$. Fix $i \in [d]$ and assume that the largest eigenvalue of A is well-separated from the rest of the spectrum, that is, $\lambda_1(A) - \lambda_2(A) \geq \delta > 0$. Then the angle θ between the top eigenvectors $v_1(A)$ and $v_1(B)$ of A and B, respectively, satisfies $\sin \theta \leq 2 \|A - B\|_2 / \delta$.*

As an easy corollary, it implies that the top eigenvectors $v_1(A)$ and $v_1(B)$ are close to each other up to a sign, namely, there exists $s \in \{-1, 1\}$ such that

$$\|v_1(A) - s \, v_1(B)\|_2 \leq \frac{2^{3/2} \|A - B\|_2}{\delta} .$$

REFERENCES

[1] Aditya Bhaskara and Aravindan Vijayaraghavan. 2011. Approximating Matrix P-norms. In *Proceedings of the Twenty-second Annual ACM-SIAM Symposium on Discrete Algorithms* (San Francisco, California) *(SODA '11)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 497–511.

[2] Vijay Bhattiprolu, Mrinalkanti Ghosh, Venkatesan Guruswami, Euiwoong Lee, and Madhur Tulsiani. 2019. Approximability of $P \rightarrow Q$ Matrix Norms: Generalized Krivine Rounding and Hypercontractive Hardness. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms* (San Diego, California) *(SODA '19)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1358–1368.

[3] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. *In International Conference on Learning Representations* (2015).

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

[5] Valentin Khrulkov and Ivan Oseledets. 2018. Art of Singular Vectors and Universal Adversarial Perturbations. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[6] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2017. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533* (2017).

[7] Aleksander Madry, Aleksandar A Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. *In International Conference on Learning Representations* (2018).

[8] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, Pascal Frossard, and Stefano Soatto. 2017. Analysis of universal adversarial perturbations. *CoRR* abs/1705.09554 (2017). arXiv:1705.09554

[9] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. 2017. Universal adversarial perturbations. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017).

[10] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. 2016. DeepFool: a simple and accurate method to fool deep neural networks. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016).

[11] Konda Reddy Mopuri, Utsav Garg, and R Venkatesh Babu. 2017. Fast Feature Fool: A data independent approach to universal adversarial perturbations. In *Proceedings of the British Machine Vision Conference (BMVC)*.

[12] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision* 115, 3 (01 Dec 2015), 211–252. https://doi.org/10.1007/s11263-015-0816-y

[13] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. (2013). arXiv:1312.6199 [cs.CV]

[14] Florian Tramer, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick Mc-Daniel. 2017. The Space of Transferable Adversarial Examples. *arXiv preprint arXiv:1704.03453* (2017).

[15] Roman Vershynin. 2018. *High-Dimensional Probability: An Introduction with Applications in Data Science.* Cambridge University Press.