Sparse-RS: A Versatile Framework for Query-Efficient Sparse Black-Box Adversarial Attacks

Francesco Croce,[1] Maksym Andriushchenko,[2] Naman D. Singh,[1] Nicolas Flammarion,[2] Matthias Hein[1]

[1] University of Tübingen
[2] EPFL

Abstract

We propose a versatile framework based on random search, `Sparse-RS`, for score-based sparse targeted and untargeted attacks in the black-box setting. `Sparse-RS` does not rely on substitute models and achieves state-of-the-art success rate and query efficiency for multiple sparse attack models: l_0-bounded perturbations, adversarial patches, and adversarial frames. The l_0-version of untargeted `Sparse-RS` outperforms all black-box and even all white-box attacks for different models on MNIST, CIFAR-10, and ImageNet. Moreover, our untargeted `Sparse-RS` achieves very high success rates even for the challenging settings of 20×20 adversarial patches and 2-pixel wide adversarial frames for 224×224 images. Finally, we show that `Sparse-RS` can be applied to generate targeted universal adversarial patches where it significantly outperforms the existing approaches. Our code is available at https://github.com/fra31/sparse-rs.

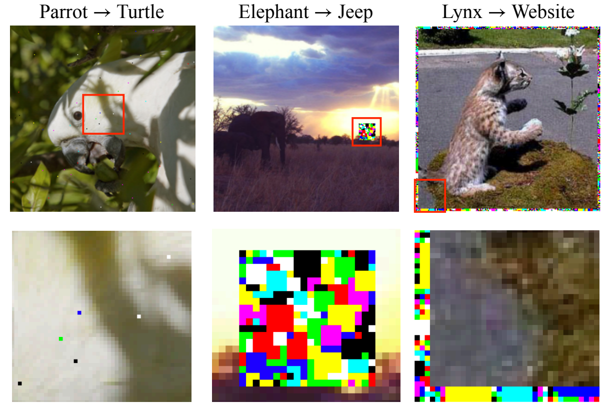

Figure 1: Adversarial examples for sparse threat models (l_0-bounded, patches, frames) generated with our black-box `Sparse-RS` framework which does not require surrogate models and is more query efficient.

Introduction

The discovery of the vulnerability of neural networks to adversarial examples (Biggio et al. 2013; Szegedy et al. 2014) revealed that the decision of a classifier or a detector can be changed by small, carefully chosen perturbations of the input. Many efforts have been put into developing increasingly more sophisticated attacks to craft small, semantics-preserving modifications which are able to fool classifiers and bypass many defense mechanisms (Carlini and Wagner 2017; Athalye, Carlini, and Wagner 2018). This is typically achieved by constraining or minimizing the l_p-norm of the perturbations, usually either l_∞ (Szegedy et al. 2014; Kurakin, Goodfellow, and Bengio 2017; Carlini and Wagner 2017; Madry et al. 2018; Croce and Hein 2020), l_2 (Moosavi-Dezfooli, Fawzi, and Frossard 2016; Carlini and Wagner 2017; Rony et al. 2019; Croce and Hein 2020) or l_1 (Chen et al. 2018; Modas, Moosavi-Dezfooli, and Frossard 2019; Croce and Hein 2020). Metrics other than l_p-norms which are more aligned to human perception have been also recently used, e.g. Wasserstein distance (Wong, Schmidt, and Kolter 2019; Hu et al. 2020) or neural-network based ones such as LPIPS (Zhang et al. 2018; Laidlaw, Singla, and Feizi 2021). All these attacks have in common the tendency to modify all the elements of the input.

Conversely, *sparse attacks* pursue an opposite strategy: they perturb only a small portion of the original input but possibly with large modifications. Thus, the perturbations are indeed visible but do not alter the semantic content, and can even be applied in the physical world (Lee and Kolter 2019; Thys, Van Ranst, and Goedemé 2019; Li, Schmidt, and Kolter 2019). Sparse attacks include l_0-attacks (Narodytska and Kasiviswanathan 2017; Carlini and Wagner 2017; Papernot et al. 2016; Schott et al. 2019; Croce and Hein 2019), adversarial patches (Brown et al. 2017; Karmon, Zoran, and Goldberg 2018; Lee and Kolter 2019) and frames (Zajac et al. 2019), where the perturbations have some predetermined structure. Moreover, sparse attacks generalize to tasks outside computer vision, such as malware detection or natural language processing, where the nature of the domain imposes to modify only a limited number of input features (Grosse et al. 2016; Jin et al. 2019).

We focus on the black-box score-based scenario, where the attacker can only access the predicted scores of a classifier f, but does not know the network weights and in particularly cannot use gradients of f wrt the input (as in the white-box setup). We do not consider more restrictive (e.g., decision-based attacks (Brendel, Rauber, and Bethge 2018;

Brunner et al. 2019) where the adversary only knows the label assigned to each input) or more permissive (e.g., a surrogate model similar to the victim one is available (Cheng et al. 2019; Huang and Zhang 2020)) cases. For the l_0-threat model only a few black-box attacks exist (Narodytska and Kasiviswanathan 2017; Schott et al. 2019; Croce and Hein 2019; Zhao et al. 2019), which however do not focus on query efficiency or scale to datasets like ImageNet without suffering from prohibitive computational cost. For adversarial patches and frames, black-box methods are mostly limited to transfer attacks, that is a white-box attack is performed on a surrogate model, with the exception of (Yang et al. 2020) who use a predefined dictionary of patches.

Contributions. Random search is particularly suitable for zeroth-order optimization in presence of complicated combinatorial constraints, as those of sparse threat models. Then, we design specific sampling distributions for the random search algorithm to efficiently generate sparse black-box attacks. The resulting `Sparse-RS` is a simple and flexible framework which handles

- **l_0-perturbations**: `Sparse-RS` significantly outperforms the existing black-box attacks in terms of the query efficiency and success rate, and leads to a better success rate even when compared to the state-of-the-art *white-box* attacks on standard and robust models.
- **Adversarial patches**: `Sparse-RS` achieves better results than both TPA (Yang et al. 2020) and a black-box adaptations of projected gradient descent (PGD) attacks via gradient estimation.
- **Adversarial frames**: `Sparse-RS` outperforms the existing adversarial framing method (Zajac et al. 2019) with gradient estimation and achieves a very high success rate even with 2-pixel wide frames.

Due to space reasons the results for adversarial frames had to be moved to the appendix, available in the extended version at https://arxiv.org/abs/2006.12834.

Black-box Adversarial Attacks

Let $f : \mathcal{S} \subseteq \mathbb{R}^d \rightarrow \mathbb{R}^K$ be a classifier which assigns input $x \in \mathcal{S}$ to class $y = \arg\max_{r=1,\ldots,K} f_r(x)$. The goal of an *untargeted* attack is to craft a perturbation $\delta \in \mathbb{R}^d$ s.t.

$$\operatorname*{arg\,max}_{r=1,\ldots,K} f_r(x+\delta) \neq y, \quad x+\delta \in \mathcal{S} \quad \text{and} \quad \delta \in \mathcal{T},$$

where $\mathcal{S}$ is the input domain and $\mathcal{T}$ are the constraints the adversarial perturbation has to fulfill (e.g. bounded l_p-norm), while a *targeted* attack aims at finding δ such that

$$\operatorname*{arg\,max}_{r=1,\ldots,K} f_r(x+\delta) = t, \quad x+\delta \in \mathcal{S} \quad \text{and} \quad \delta \in \mathcal{T},$$

with t as target class. Generating such δ can be translated into an optimization problem as

$$\min_{\delta \in \mathbb{R}^d} \; L(f(x+\delta), t) \quad \text{s.t.} \quad x+\delta \in \mathcal{S} \quad \text{and} \quad \delta \in \mathcal{T} \quad (1)$$

by choosing a label t and loss function L whose minimization leads to the desired classification. By *threat model* we mean the overall attack setting determined by the goal of the attacker (targeted vs untargeted attack), the level of knowledge (white- vs black-box), and the perturbation set $\mathcal{T}$.

Many algorithms have been proposed to solve Problem (1) in the black-box setting where one cannot use gradient-based methods. One of the first approaches is by (Fawzi and Frossard 2016) who propose to sample candidate adversarial occlusions via the Metropolis MCMC method, which can be seen as a way to generate adversarial patches whose content is not optimized. (Ilyas et al. 2018; Uesato et al. 2018) propose to approximate the gradient through finite difference methods, later improved to reduce their high computational cost in terms of queries of the victim models (Bhagoji et al. 2018; Tu et al. 2019; Ilyas, Engstrom, and Madry 2019). Alternatively, (Alzantot et al. 2019; Liu et al. 2019) use genetic algorithms in the context of image classification and malware detection respectively. A line of research has focused on rephrasing l_∞-attacks as discrete optimization problems (Moon, An, and Song 2019; Al-Dujaili and O'Reilly 2020; Meunier, Atif, and Teytaud 2019), where specific techniques lead to significantly better query efficiency. (Guo et al. 2019) adopt a variant of random search to produce perturbations with a small l_2-norm.

Closest in spirit is the Square Attack of (Andriushchenko et al. 2020), which is state-of-the-art for l_∞- and l_2-bounded black-box attacks. It uses random search to iteratively generate samples on the surface of the l_∞- or l_2-ball. Together with a particular sampling distribution based on square-shaped updates and a specific initialization, this leads to a simple algorithm which outperforms more sophisticated attacks in success rate and query efficiency. In this paper we show that the random search idea is ideally suited for sparse attacks, where the non-convex, combinatorial constraints are not easily handled even by gradient-based *white-box* attacks.

`Sparse-RS` Framework

Random search (RS) is a well known scheme for derivative free optimization (Rastrigin 1963). Given an objective function L to minimize, a starting point $x^{(0)}$ and a sampling distribution $\mathcal{D}$, an iteration of RS at step i is given by

$$\delta \sim \mathcal{D}(x^{(i)}), \quad x^{(i+1)} = \operatorname*{arg\,min}_{y \in \{x^{(i)},\, x^{(i)}+\delta\}} L(y). \qquad (2)$$

At every step an update of the current iterate $x^{(i)}$ is sampled according to $\mathcal{D}$ and accepted only if it decreases the objective value, otherwise the procedure is repeated. Although not explicitly mentioned in Eq. (2), constraints on the iterates $x^{(i)}$ can be integrated by ensuring that δ is sampled so that $x^{(i)} + \delta$ is a feasible solution. Thus even complex, e.g. combinatorial, constraints can easily be integrated as RS just needs to be able to produce feasible points in contrast to gradient-based methods which depend on a continuous set to optimize over. While simple and flexible, RS is an effective tool in many tasks (Zabinsky 2010; Andriushchenko et al. 2020), with the key ingredient for its success being a task-specific sampling distribution $\mathcal{D}$ to guide the exploration of the space of possible solutions.

We summarize our general framework based on random search to generate sparse adversarial attacks, `Sparse-RS`,

Algorithm 1: `Sparse-RS`

input : loss L, input x_{orig}, max query N, sparsity k, input space constraints $\mathcal{S}$
output: approximate minimizer of L

1 $M \leftarrow k$ indices of elements to be perturbed
2 $\Delta \leftarrow$ values of the perturbation to be applied
3 $z \leftarrow x_{\text{orig}}, \quad z_M \leftarrow \Delta$ `// set elements in` M `to values in` Δ
4 $L^* \leftarrow L(z), \quad i \leftarrow 0$ `// initialize loss`
5 **while** $i < N$ **and** *success not achieved* **do**
6 $\quad M' \leftarrow$ sampled modification of M `// new set of indices`
7 $\quad \Delta' \leftarrow$ sampled modification of Δ `// new perturbation`
8 $\quad z \leftarrow x_{\text{orig}}, \quad z_{M'} \leftarrow \Delta'$ `// create new candidate in` $\mathcal{S}$
9 $\quad$ **if** $L(z) < L^*$ **then**
10 $\quad\quad L^* \leftarrow L(z), \ M \leftarrow M', \ \Delta \leftarrow \Delta'$ `// if loss improves, update sets`
11 $\quad i \leftarrow i + 1$
12 $z \leftarrow x_{\text{orig}}, \quad z_M \leftarrow \Delta$ `// return best` Δ
13 **return** z

in Alg. 1, where the sparsity k indicates the maximum number of features that can be perturbed. A sparse attack is characterized by two variables: the set of components to be perturbed M and the values Δ to be inserted at M to form the adversarial input. To optimize over both of them we first sample a random update of the locations M of the current perturbation (step 6) and then a random update of its values Δ (step 7). In some threat models (e.g. adversarial frames) the set M cannot be changed, so $M' \equiv M$ at every step. How Δ' is generated depends on the specific threat model, so we present the individual procedures in the next sections. We note that for all threat models, the runtime is dominated by the cost of a forward pass through the network, and all other operations are computationally inexpensive.

Common to all variants of `Sparse-RS` is that the whole budget for the perturbations is fully exploited both in terms of number of modified components and magnitude of the elementwise changes (constrained only by the limits of the input domain $\mathcal{S}$). This follows the intuition that larger perturbations should lead faster to an adversarial example. Moreover, the difference of the candidates M' and Δ' with M and Δ shrinks gradually with the iterations which mimics the reduction of the step size in gradient-based optimization: initial large steps allow to quickly decrease the objective loss, but smaller steps are necessary to refine a close-to-optimal solution at the end of the algorithm. Finally, we impose a limit N on the maximum number of queries of the classifier, i.e. evaluations of the objective function.

As objective function L to be minimized, we use in the case of untargeted attacks the margin loss $L_{\text{margin}}(f(\cdot), y) = f_y(\cdot) - \max_{r \neq y} f_r(\cdot)$, where y is the correct class, so that $L < 0$ is equivalent to misclassification, whereas for targeted attacks we use the cross-entropy loss L_{CE} of the target class t, namely $L_{\text{CE}}(f(\cdot), t) = -f_t(\cdot) + \log\left(\sum_{r=1}^{K} e^{f_r(\cdot)}\right)$.

The code of the `Sparse-RS` framework is available at https://github.com/fra31/sparse-rs.

`Sparse-RS` for l_0-Bounded Attacks

The first threat model we consider are l_0-bounded adversarial examples where only up to k pixels or k features/color channels of an input $x_{\text{orig}} \in [0,1]^{h \times w \times c}$ (width w, height h, color c) can be modified, but there are no constraints on the magnitude of the perturbations except for those of the input domain. Note that constraining the number of perturbed pixels or features leads to two different threat models which are not directly comparable. Due to the combinatorial nature of the l_0-threat model, this turns out to be quite difficult for continuous optimization techniques which are more prone to get stuck in suboptimal maxima.

l_0-`RS` algorithm. We first consider the threat model where up to k pixels can be modified. Let U be the set of the $h \cdot w$ pixels. In this case the set $M \subset U$ from Alg. 1 is initialized sampling uniformly k elements of U, while $\Delta \sim \mathcal{U}(\{0,1\}^{k \times c})$, that is random values in $\{0,1\}$ (every perturbed pixel gets one of the corners of the color cube $[0,1]^c$). Then, at the i-th iteration, we randomly select $A \subset M$ and $B \subset U \setminus M$, with $|A| = |B| = \alpha^{(i)} \cdot k$, and create $M' = (M \setminus A) \cup B$. Δ' is formed by sampling random values from $\{0,1\}^c$ for the elements in B, i.e. those which were not perturbed at the previous iteration. The quantity $\alpha^{(i)}$ controls how much M' differs from M and decays following a predetermined piecewise constant schedule rescaled according to the maximum number of queries N. The schedule is completely determined by the single value α_{init}, used to calculate $\alpha^{(i)}$ for every iteration i, which is also the only free hyperparameter of our scheme. We provide details about the algorithm, schedule, and values of α_{init} in App. A and B, and ablation studies for them in App. G. For the feature based threat model each color channel is treated as a pixel and one applies the scheme above to the "grayscale" image ($c = 1$) with three times as many "pixels".

Comparison of Query Efficiency of l_0-RS

We compare pixel-based l_0-RS to other black-box untargeted attacks in terms of success rate versus query efficiency. The results of targeted attacks are in App. A. Here we focus on attacking normally trained VGG-16-BN and ResNet-50 models on ImageNet, which contains RGB images resized to shape 224×224, that is 50,176 pixels, belonging to 1,000 classes. We consider perturbations of size $k \in \{50, 150\}$ pixels to assess the effectiveness of the untargeted attacks at different thresholds with a limit of 10,000 queries. We evaluate the success rate on the initially correctly classified images out of 500 images from the validation set.

Competitors. Many existing black-box pixel-based l_0-attacks (Narodytska and Kasiviswanathan 2017; Schott et al. 2019; Croce and Hein 2019) do not aim at query efficiency and rather try to minimize the size of the perturbations. Among them, only CornerSearch (Croce and Hein 2019) and ADMM attack (Zhao et al. 2019) scale to ImageNet. However, CornerSearch requires $8 \times \#pixels$ queries only for the

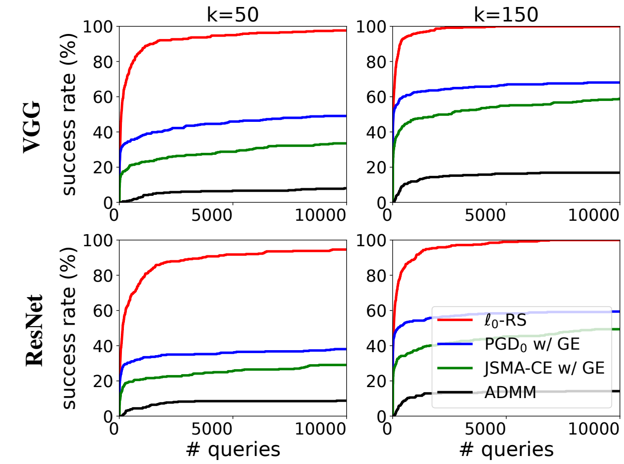

Figure 2: Progression of the success rate vs number of queries for black-box pixel-based l_0-attacks on ImageNet in the untargeted setting. At all sparsity levels l_0−RS (red) outperforms PGD$_0$ (blue) and JSMA-CE (green) with gradient estimation and ADMM attack (black).

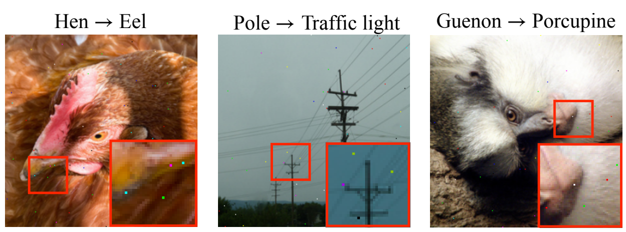

Figure 3: Untargeted l_0-adversarial examples generated by our pixel-based l_0−RS algorithm for $k = 50$ pixels.

initial phase, exceeding the query limit we fix by more than 40 times. The ADMM attack tries to achieve a successful perturbation and then reduces its l_0-norm. Moreover, we introduce black-box versions of PGD$_0$ (Croce and Hein 2019) with the gradient estimated by finite difference approximation as done in prior work, e.g., see (Ilyas et al. 2018). As a strong baseline, we introduce JSMA-CE which is a version of the JSMA algorithm (Papernot et al. 2016) that we adapt to the black-box setting: (1) for query efficiency, we estimate the gradient of the cross-entropy loss instead of gradients of *each* class logit, (2) on each iteration, we modify the *pixels* with the highest gradient contribution. More details about the attacks can be found in App. A.

Results. We show in Fig. 2 the success rate vs the number of queries for all black-box attacks. In all cases, l_0−RS outperforms its competitors in terms of the final success rate by a large margin—the second best method (PGD$_0$ w/ GE) is at least 30% worse. Moreover, l_0−RS is query efficient as it achieves results close to the final ones already with a low number of queries. For example, on VGG with $k = 150$, l_0−RS achieves 100% of success rate using on average only 171 queries, with a median of 25. Unlike other methods, l_0−RS can achieve almost 100% success rate by perturbing

attack	*type*	VGG	ResNet
l_0-bound in **pixel space** $k = 50$			
JSMA-CE	white-box	42.6%	39.6%
PGD$_0$	white-box	87.0%	81.2%
ADMM	black-box	30.3%	29.0%
JSMA-CE with GE	black-box	49.6%	44.8%
PGD$_0$ with GE	black-box	61.4%	51.8%
CornerSearch*	black-box	82.0%	72.0%
l_0−RS	black-box	**98.2%**	**95.8%**
l_0-bound in **feature space** $k = 50$			
SAPF*	white-box	21.0%	18.0%
ProxLogBarrier	white-box	33.0%	28.4%
EAD	white-box	39.8%	35.6%
SparseFool	white-box	43.6%	42.0%
VFGA	white-box	58.8%	55.2%
FMN	white-box	83.8%	77.6%
PDPGD	white-box	89.6%	87.2%
ADMM	black-box	32.6%	29.0%
CornerSearch*	black-box	76.0%	62.0%
l_0−RS	black-box	**92.8%**	**88.8%**

Table 1: ImageNet: Robust test error of l_0-attacks. The entries with * are evaluated on 100 points instead of 500 because of their high computational cost. All black-box attacks use 10k queries except CornerSearch which uses 600k. l_0−RS outperforms all black- *and* white-box attacks.

50 pixels which is *only* 0.1% of the total number of pixels. We visualize the adversarial examples of l_0−RS in Fig. 3.

Using l_0-RS for Accurate Robustness Evaluation

In this section, our focus is the accurate evaluation of robustness in the l_0-threat model. For this, we evaluate existing white-box methods and black-box methods together. Instead of the success rate taken only over correctly classified examples, here we rather consider *robust error* (similarly to (Madry et al. 2018)), which is defined as the classification error on the adversarial examples crafted by an attacker.

White-box attacks on ImageNet. We test the robustness of the ImageNet models introduced in the previous section to l_0-bounded perturbations. As competitors we consider multiple white-box attacks which minimize the l_0-norm in *feature space*: SAPF (Fan et al. 2020), ProxLogBarrier (Pooladian et al. 2020), EAD (Chen et al. 2018), SparseFool (Modas, Moosavi-Dezfooli, and Frossard 2019), VFGA (Césaire et al. 2020), FMN (Pintor et al. 2021) and PDPGD (Matyasko and Chau 2021). For the l_0-threat model in *pixel space* we use two white-box baselines: PGD$_0$ (Croce and Hein 2019), and JSMA-CE (Papernot et al. 2016) (where we use the gradient of the cross-entropy loss to generate the saliency map). Moreover, we show the results of the black-box attacks from the previous section (all with a query limit of 10,000), and additionally use the black-box CornerSearch for which we use a query limit of 600k and which is thus only partially comparable. Details of the attacks are available in App. A. Table 1 shows the robust error given by all competitors: l_0−RS achieves the best results for

attack	*type*	l_2-AT ResNet	l_1-AT ResNet
	l_0-bound in **pixel space** $k = 24$		
PGD$_0$	white-box	68.7%	72.7%
CornerSearch	black-box	59.3%	64.9%
l_0−RS	black-box	**85.7%**	**81.0%**
	l_0-bound in **feature space** $k = 24$		
VFGA	white-box	40.5%	27.5%
FMN	white-box	52.9%	28.2%
PDPGD	white-box	46.4%	26.9%
CornerSearch	black-box	43.2%	29.4%
l_0−RS	black-box	**63.4%**	**38.3%**

Table 2: CIFAR10: Robust test error of untargeted l_0-attacks in pixel and feature space on a l_2- resp. l_1-AT model.

pixel and feature based l_0-threat model on both VGG and ResNet, outperforming black- *and* white-box attacks. We note that while the PGD attack has been observed to give accurate robustness estimates for l_∞- and l_2-norms (Madry et al. 2018), this is not the case for the l_0 constraint set. This is due to the discrete structure of the l_0-ball which is not amenable for continuous optimization.

Comparison on CIFAR-10. In Table 2 we compare the strongest white- and black-box attacks on l_1- resp. l_2-adversarially trained PreAct ResNet-18 on CIFAR-10 from (Croce and Hein 2021) and (Rebuffi et al. 2021) (details in App. A.5). We keep the same computational budget used on ImageNet. As before, we consider perturbations with l_0-norm $k = 24$ in pixel or feature space: in both cases l_0−RS achieves the highest robust test error outperforming even all white-box attacks. Note that, as expected, the model robust wrt l_1 is less vulnerable to l_0-attacks especially in the feature space, whose l_1-norm is close to that used during training.

Robust generative models on MNIST. (Schott et al. 2019) propose two robust generative models on MNIST, ABS and Binary ABS, which showed high robustness against multiple types of l_p-bounded adversarial examples. These classifiers rely on optimization-based inference using a variational auto-encoder (VAE) with 50 steps of gradient descent for each prediction (times 1,000 repetitions). It is too expensive to get gradients with respect to the input through the optimization process, thus (Schott et al. 2019) evaluate only black-box attacks, and test l_0-robustness with sparsity $k = 12$ using their proposed Pointwise Attack with 10 restarts. We evaluate on both models CornerSearch with a budget of 50,000 queries and l_0−RS with an equivalent budget of 10,000 queries and 5 random restarts. Table 3 summarizes the robust test error (on 100 points) achieved by the attacks (the results of Pointwise Attack are taken from (Schott et al. 2019)). For both classifiers, l_0-RS yields the strongest evaluation of robustness suggesting that the ABS models are less robust than previously believed. This illustrates that despite we have *full access* to the attacked VAE model, a strong *black-box* l_0-attack can still be useful for an accurate robustness evaluation.

l_0−RS on malware detection. We apply our method on a malware detection task and show its effectiveness in App. B.

attack	*type*	$k = 12$ (pixels) ABS	Binary ABS
Pointwise Attack	black-box	31%	23%
CornerSearch	black-box	29%	28%
l_0−RS	black-box	**55%**	**51%**

Table 3: Robust test error on robust models (Schott et al. 2019) on MNIST by different attacks on 100 test points.

Theoretical Analysis of l_0-RS

Given the empirical success of l_0−RS, here we analyze it theoretically for a binary classifier. While the analysis does not directly transfer to neural networks, most modern neural network architectures result in piecewise linear classifiers (Arora et al. 2018) , so that the result approximately holds in a sufficiently small neighborhood of the target point x.

As in the malware detection task in App. B, we assume that the input x has binary features, $x \in \{0, 1\}^d$, and we denote the label by $y \in \{-1, 1\}$ and the gradient of the linear model by $w_x \in \mathbb{R}^d$. Then the Problem (1) of finding the optimal l_0 adversarial example is equivalent to:

$$\begin{aligned}\operatorname*{arg\,min}_{\substack{\|\delta\|_0 \le k \\ x_i+\delta_i \in \{0,1\}}} y\,\langle w_x, x+\delta\rangle &= \operatorname*{arg\,min}_{\substack{\|\delta\|_0 \le k \\ \delta_i \in \{0,1-2x_i\}}} \langle y w_x, \delta\rangle \\ &= \operatorname*{arg\,min}_{\substack{\|\delta\|_0 \le k \\ \delta_i \in \{0,1\}}} \langle \underbrace{y w_x \odot (1-2x)}_{\hat{w}_x}, \delta\rangle,\end{aligned}$$

where $\odot$ denotes the elementwise product. In the white-box case, i.e. when w_x is known, the solution is to simply set $\delta_i = 1$ for the k smallest weights of $\hat{w}_x$. The black-box case, where w_x is unknown and we are only allowed to query the model predictions $\langle \hat{w}_x, z\rangle$ for any $z \in \mathbb{R}^d$, is more complicated since the naive weight estimation algorithm requires $O(d)$ queries to first estimate $\hat{w}_x$ and then to perform the attack by selecting the k minimal weights. This naive approach is prohibitively expensive for high-dimensional datasets (e.g., $d = 150{,}528$ on ImageNet assuming $224 \times 224 \times 3$ images). However, the problem of generating adversarial examples does not have to be always solved exactly, and often it is enough to find an approximate solution. Therefore we can be satisfied with only identifying k among the m smallest weights. Indeed, the focus is not on exactly identifying the solution but rather on having an algorithm that in expectation requires a *sublinear* number of queries. With this goal, we show that l_0−RS satisfies this requirement for large enough m.

Proposition 1 *The expected number t_k of queries needed for l_0−RS with $\alpha^{(i)} = 1/k$ to find a set of k weights out of the smallest m weights of a linear model is:*

$$\mathbb{E}\left[t_k\right] = \sum_{i=0}^{k-1} \frac{(d-k)k}{(k-i)(m-i)} < (d-k)k\frac{\ln(k)+2}{m-k}.$$

The proof is deferred to the supplement and resembles that of the coupon collector problem. For non-linear models, l_0−RS uses $\alpha^{(i)} > 1/k$ for better exploration initially, but then progressively reduces it. The main conclusion from

		attack	VGG *success rate*	VGG *mean queries*	VGG *med. queries*	ResNet *success rate*	ResNet *mean queries*	ResNet *med. queries*
untarget. 20×20	black-box	LOAP w/ GE	55.1% ± 0.6	5879 ± 51	7230 ± 377	40.6% ± 0.1	6870 ± 10	10000 ± 0
		TPA	46.1% ± 1.1	6085^{*} ± 83	8080^{*} ± 1246	49.0% ± 1.2	5722^{*} ± 64	5280^{*} ± 593
		`Sparse-RS` + SH	82.6%	2479	514	75.3%	3290	549
		`Sparse-RS` + SA	85.6% ± 1.1	2367 ± 83	533 ± 40	78.5% ± 1.0	2877 ± 64	458 ± 43
		`Patch-RS`	**87.8% ± 0.7**	**2160 ± 44**	**429 ± 22**	**79.5% ± 1.4**	**2808 ± 89**	**438 ± 68**
		White-box LOAP	98.3%	-	-	82.2%	-	-
targeted 40×40	black-box	LOAP w/ GE	23.9% ± 0.9	44134 ± 71	50000 ± 0	18.4% ± 0.9	45370 ± 88	50000 ± 0
		TPA	5.1% ± 1.2	29934^{*} ± 462	34000^{*} ± 0	6.0% ± 0.5	31690^{*} ± 494	34667^{*} ± 577
		`Sparse-RS` + SH	63.6%	25634	19026	48.6%	31250	50000
		`Sparse-RS` + SA	70.9% ± 1.2	23749 ± 346	15569 ± 568	53.7% ± 0.9	32290 ± 239	40122 ± 2038
		`Patch-RS`	**72.7% ± 0.9**	**22912 ± 207**	**14407 ± 866**	**55.6% ± 1.5**	**30290 ± 317**	**34775 ± 2660**
		White-box LOAP	99.4%	-	-	94.8%	-	-

Table 4: Success rate and query statistics of image-specific patches. Black-box attacks are given 10k/50k queries for untargeted/targeted case. SH is a deterministic method. The query statistics are computed on *all* images with 5 random seeds. * TPA uses an early stopping mechanism to save queries, thus might not use all queries. `Patch-RS` outperforms all other methods in success rate and query efficiency.

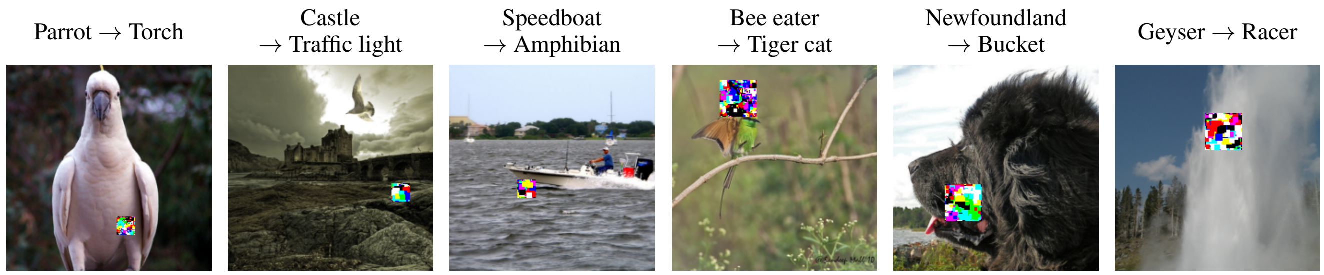

Figure 4: Image-specific untargeted (20×20 pixels, left) and targeted (40×40, right) patches generated by `Patch-RS`.

Proposition 1 is that $\mathbb{E}[t_k]$ becomes sublinear for large enough gap $m - k$, as we illustrate in Fig. 11 in App. F.

`Sparse-RS` for Adversarial Patches

Another type of sparse attacks which recently received attention are adversarial patches introduced by (Brown et al. 2017). There the perturbed pixels are localized, often square-shaped, and limited to a small portion of the image but can be changed in an arbitrary way. While some works (Brown et al. 2017; Karmon, Zoran, and Goldberg 2018) aim at universal patches, which fool the classifier regardless of the image and the position where they are applied, which we consider in the next section, we focus first on image-specific patches as in (Yang et al. 2020; Rao, Stutz, and Schiele 2020) where one optimizes both the content and the location of the patch for each image independently.

General algorithm for patches. Note that both location (step 6 in Alg. 1) and content (step 7 in Alg. 1) have to be optimized in `Sparse-RS`, and on each iteration we check only one of these updates. We test the effect of different frequencies of location/patch updates in an ablation study in App. G.2. Since the location of the patch is a discrete variable, random search is particularly well suited for its optimization. For the location updates in step 6 in Alg. 1, we randomly sample a new location in a 2D l_∞-ball around the current patch position (using clipping so that the patch is fully contained in the image). The radius of this l_∞-ball shrinks with increasing iterations in order to perform progressively more local optimization (see App. C for details).

For the update of the patch itself in step 7 in Alg. 1, the only constraints are given by the input domain $[0, 1]^d$. Thus in principle any black-box method for an l_∞-threat model can be plugged in there. We use Square Attack (SA) (Andriushchenko et al. 2020) and SignHunter (SH) (Al-Dujaili and O'Reilly 2020) as they represent the state-of-the-art in terms of success rate and query efficiency. We integrate both in our framework and refer to them as `Sparse-RS` + SH and `Sparse-RS` + SA. Next we propose a novel random search based attack motivated by SA which together with our location update yields our novel `Patch-RS` attack.

`Patch-RS`. While SA and SH are state-of-the-art for l_∞-attacks, they have been optimized for rather small perturbations whereas for patches all pixels can be manipulated arbitrarily in $[0, 1]$. Here, we design an initialization scheme and a sampling distribution specific for adversarial patches. As initialization (step 2 of Alg. 1), `Patch-RS` uses randomly placed squares with colors in $\{0, 1\}^3$, then it samples updates of the patch (step 7) with shape of squares, of size decreasing according to a piecewise constant schedule, until a refinement phase in the last iterations, when it performs single-channel updates (exact schedule in App. C). This is in contrast to SA where random vertical stripes are used as initialization and always updates for all three channels of a pixel are sampled. The ablation study in App. G.2 shows how both modifications contribute to the improved performance of `Patch-RS`.

Experiments. In addition to `Sparse-RS` + SH, `Sparse-RS` + SA, and `Patch-RS`, we consider two existing methods. i) TPA (Yang et al. 2020) which is a black-box attack aiming to produce image-specific adv. patches based on reinforcement learning. While (Yang et al. 2020) allows multiple patches for an image, we use TPA in the standard setting of a single patch. ii) Location-Optimized Adversarial Patches (LOAP) (Rao, Stutz, and Schiele 2020), a white-box attack that uses PGD for the patch updates, which we combine with gradient estimation in order to use it in the black-box scenario (see App. C for details). In Table 4 we report success rate, mean and median number of queries used for untargeted attacks with patch size 20×20 and query limit of 10,000 and for targeted attacks (random target class for each image) with patch size 40×40 and maximally 50,000 queries. We attack 500 images of ImageNet with VGG and ResNet as target models. The query statistics are computed on *all* 500 images, i.e. without restricting to only successful adversarial examples, as this makes the query efficiency comparable for different success rates. Our `Sparse-RS` + SH, `Sparse-RS` + SA and `Patch-RS` outperform existing methods by a large margin, showing the effectiveness of our scheme to optimize both location and patch. Among them, our specifically designed `Patch-RS` achieves the best results in all metrics. We visualize its resulting adversarial examples in Fig. 4.

Universal Adversarial Patches

A challenging threat model is that of a black-box, targeted universal adversarial patch attack where the classifier should be fooled into a chosen target class when the patch is applied inside any image of some other class. Previous works rely on transfer attacks: in (Brown et al. 2017) the universal patch is created using a white-box attack on surrogate models, while the white-box attack of (Karmon, Zoran, and Goldberg 2018) directly optimizes the patch for the target model on a set of training images and then only tests generalization to unseen images. Our goal is a targeted black-box attack which crafts universal patches that generalize to unseen images when applied at random locations (see examples in Fig. 5). To our knowledge, this is the first method for this threat model which does not rely on a surrogate model.

We employ Alg. 1 where for the creation of the patches in step 7 we use either SH, SA or our novel sampling distribution introduced in `Patch-RS` in the previous section. The loss in Alg. 1 is computed on a small batch of 30 training images and the initial locations M of the patch in each of the training images are sampled randomly. In order not to overfit on the training batch, we resample training images and locations of the patches (step 6 in Alg. 1) every $10k$ queries (total query budget $100k$). As stochastic gradient descent this is a form of stochastic optimization of the population loss (expectation over images and locations) via random search.

Experiments. We apply the above scheme to `Sparse-RS` + SH/SA and `Patch-RS` to create universal patches of size 50×50 for 10 random target classes on VGG (we repeat it for 3 seeds for RS-based methods). We compare to (1) the transfer-based attacks obtained via PGD (Madry et al. 2018) and MI-FGSM (Dong et al. 2018) using ResNet as surrogate model, and to (2) ZO-AdaMM (Chen et al. 2019) based on gradient estimation. The results in Table 5 show that our `Sparse-RS` + SH/SA and `Patch-RS` outperform other methods by large margin. We provide extra details and results for frames in App. E.

attack	VGG
Transfer PGD	3.3%
Transfer MI-FGSM	1.3%
PGD w/ GE	35.1%
ZO-AdaMM	45.8%
`Sparse-RS` + SH	63.9%
`Sparse-RS` + SA	**72.9%** $\pm$ **3.6**
`Patch-RS`	70.8% $\pm$ 1.3

Table 5: Success rate of targeted universal 50×50 patches.

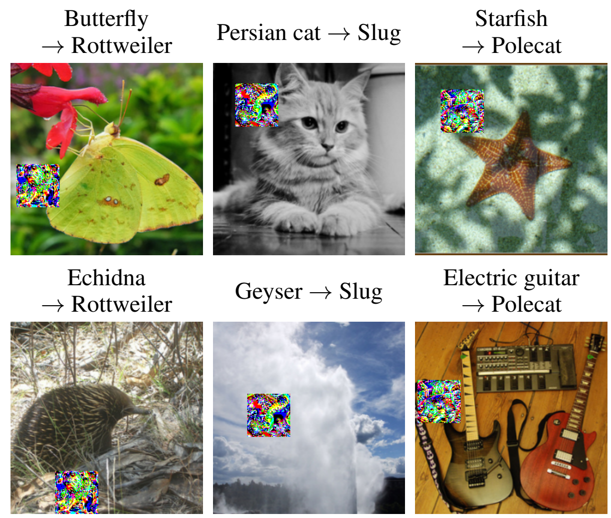

Figure 5: We visualize two images in each column with the *same* targeted universal patch generated by `Patch-RS` that changes the predictions to the desired target class.

Conclusion

We propose a versatile framework, `Sparse-RS`, which achieves state-of-the-art success rate and query efficiency in multiple sparse threat models: l_0-perturbations, adversarial patches and adversarial frames (see App. D). Moreover, it is effective in the challenging task of crafting universal adversarial patches without relying on surrogate models, unlike the existing methods. We think that strong black-box adversarial attacks are a very important component to assess the robustness against such localized and structured attacks, which go beyond the standard l_p-threat models.

Acknowledgements

We thank Yang et al. (2020) for quickly releasing their code and answering our questions. F.C., N.S. and M.H. acknowledge support from the German Federal Ministry of Education and Research (BMBF) through the Tübingen AI Center (FKZ: 01IS18039A), the DFG Cluster of Excellence "Machine Learning – New Perspectives for Science", EXC 2064/1, project number 390727645, and by DFG grant 389792660 as part of TRR 248.

References

Al-Dujaili, A.; Huang, A.; Hemberg, E.; and O'Reilly, U.-M. 2018. Adversarial deep learning for robust detection of binary encoded malware. In *IEEE S&P Workshops*, 76–82.

Al-Dujaili, A.; and O'Reilly, U.-M. 2020. There are No Bit Parts for Sign Bits in Black-Box Attacks. In *ICLR*.

Alzantot, M.; Sharma, Y.; Chakraborty, S.; and Srivastava, M. 2019. Genattack: practical black-box attacks with gradient-free optimization. In *GECCO*.

Andriushchenko, M.; Croce, F.; Flammarion, N.; and Hein, M. 2020. Square Attack: a query-efficient black-box adversarial attack via random search. In *ECCV*.

Arora, R.; Basuy, A.; Mianjyz, P.; and Mukherjee, A. 2018. Understanding deep neural networks with rectified linear unit. In *ICLR*.

Arp, D.; Spreitzenbarth, M.; Hubner, M.; Gascon, H.; Rieck, K.; and Siemens, C. 2014. Drebin: Effective and explainable detection of android malware in your pocket. In *NDSS*, volume 14, 23–26.

Athalye, A.; Carlini, N.; and Wagner, D. A. 2018. Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples. In *ICML*.

Bhagoji, A. N.; He, W.; Li, B.; and Song, D. 2018. Practical black-box attacks on deep neural networks using efficient query mechanisms. In *ECCV*.

Biggio, B.; Corona, I.; Maiorca, D.; Nelson, B.; Srndic, N.; Laskov, P.; Giacinto, G.; and Roli, F. 2013. Evasion Attacks against Machine Learning at Test Time. In *ECML/PKKD*.

Brendel, W.; Rauber, J.; and Bethge, M. 2018. Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models. In *ICLR*.

Brown, T. B.; Mané, D.; Roy, A.; Abadi, M.; and Gilmer, J. 2017. Adversarial Patch. In *NeurIPS 2017 Workshop on Machine Learning and Computer Security*.

Brunner, T.; Diehl, F.; Le, M. T.; and Knoll, A. 2019. Guessing smart: biased sampling for efficient black-box adversarial attacks. In *ICCV*.

Carlini, N.; and Wagner, D. 2017. Towards Evaluating the Robustness of Neural Networks. In *IEEE Symposium on Security and Privacy*.

Césaire, M.; Hajri, H.; Lamprier, S.; and Gallinari, P. 2020. Stochastic sparse adversarial attacks. *arXiv preprint arXiv:2011.12423*.

Chen, P.; Sharma, Y.; Zhang, H.; Yi, J.; and Hsieh, C. 2018. EAD: Elastic-Net Attacks to Deep Neural Networks via Adversarial Examples. In *AAAI*.

Chen, X.; Liu, S.; Xu, K.; Li, X.; Lin, X.; Hong, M.; and Cox, D. 2019. ZO-AdaMM: Zeroth-order adaptive momentum method for black-box optimization. In *NeurIPS*.

Cheng, S.; Dong, Y.; Pang, T.; Su, H.; and Zhu, J. 2019. Improving Black-box Adversarial Attacks with a Transfer-based Prior. In *NeurIPS*.

Croce, F.; Andriushchenko, M.; Sehwag, V.; Debenedetti, E.; Flammarion, N.; Chiang, M.; Mittal, P.; and Hein, M. 2020. RobustBench: a standardized adversarial robustness benchmark. *arXiv preprint arXiv:2010.09670*.

Croce, F.; and Hein, M. 2019. Sparse and Imperceivable Adversarial Attacks. In *ICCV*.

Croce, F.; and Hein, M. 2020. Minimally distorted Adversarial Examples with a Fast Adaptive Boundary Attack. In *ICML*.

Croce, F.; and Hein, M. 2021. Mind the box: l_1-APGD for sparse adversarial attacks on image classifiers. In *ICML*.

Dong, Y.; Liao, F.; Pang, T.; Su, H.; Zhu, J.; Hu, X.; and Li, J. 2018. Boosting Adversarial Attacks With Momentum. In *CVPR*.

Fan, Y.; Wu, B.; Li, T.; Zhang, Y.; Li, M.; Li, Z.; and Yang, Y. 2020. Sparse adversarial attack via perturbation factorization. In *ECCV*.

Fawzi, A.; and Frossard, P. 2016. Measuring the effect of nuisance variables on classifiers. In *BMVC*.

Grosse, K.; Papernot, N.; Manoharan, P.; Backes, M.; and McDaniel, P. 2016. Adversarial perturbations against deep neural networks for malware classification. *arXiv preprint arXiv:1606.04435*.

Guo, C.; Gardner, J. R.; You, Y.; Wilson, A. G.; and Weinberger, K. Q. 2019. Simple black-box adversarial attacks. In *ICML*.

Hu, J. E.; Swaminathan, A.; Salman, H.; and Yang, G. 2020. Improved Image Wasserstein Attacks and Defenses. *arXiv preprint arXiv:2004.12478*.

Hu, W.; and Tan, Y. 2017. Generating adversarial malware examples for black-box attacks based on GAN. *arXiv preprint arXiv:1702.05983*.

Huang, Z.; and Zhang, T. 2020. Black-Box Adversarial Attack with Transferable Model-based Embedding. In *ICLR*.

Ilyas, A.; Engstrom, L.; Athalye, A.; and Lin, J. 2018. Black-box adversarial attacks with limited queries and information. In *ICML*.

Ilyas, A.; Engstrom, L.; and Madry, A. 2019. Prior convictions: Black-box adversarial attacks with bandits and priors. In *ICLR*.

Jin, D.; Jin, Z.; Zhou, J. T.; and Szolovits, P. 2019. Is BERT Really Robust? Natural Language Attack on Text Classification and Entailment. In *AAAI*.

Karmon, D.; Zoran, D.; and Goldberg, Y. 2018. Lavan: Localized and visible adversarial noise. In *ICML*.

Kurakin, A.; Goodfellow, I. J.; and Bengio, S. 2017. Adversarial examples in the physical world. In *ICLR Workshop*.

Laidlaw, C.; Singla, S.; and Feizi, S. 2021. Perceptual Adversarial Robustness: Defense Against Unseen Threat Models. In *ICLR*.

Lee, M.; and Kolter, Z. 2019. On physical adversarial patches for object detection. *ICML Workshop on Security and Privacy of Machine Learning*.

Li, J.; Schmidt, F.; and Kolter, Z. 2019. Adversarial camera stickers: A physical camera-based attack on deep learning systems. In *ICML*, 3896–3904.

Liu, X.; Du, X.; Zhang, X.; Zhu, Q.; Wang, H.; and Guizani, M. 2019. Adversarial samples on android malware detection systems for IoT systems. *Sensors*, 19(4): 974.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Valdu, A. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *ICLR*.

Matyasko, A.; and Chau, L.-P. 2021. PDPGD: Primal-Dual Proximal Gradient Descent Adversarial Attack. *arXiv preprint arXiv:2106.01538*.

Meunier, L.; Atif, J.; and Teytaud, O. 2019. Yet another but more efficient black-box adversarial attack: tiling and evolution strategies. *arXiv preprint, arXiv:1910.02244*.

Modas, A.; Moosavi-Dezfooli, S.; and Frossard, P. 2019. SparseFool: a few pixels make a big difference. In *CVPR*.

Moon, S.; An, G.; and Song, H. O. 2019. Parsimonious Black-Box Adversarial Attacks via Efficient Combinatorial Optimization. In *ICML*.

Moosavi-Dezfooli, S.-M.; Fawzi, A.; and Frossard, P. 2016. DeepFool: a simple and accurate method to fool deep neural networks. In *CVPR*, 2574–2582.

Narodytska, N.; and Kasiviswanathan, S. 2017. Simple black-box adversarial attacks on deep neural networks. In *CVPR Workshops*.

Papernot, N.; McDaniel, P.; Jha, S.; Fredrikson, M.; Celik, Z. B.; and Swami, A. 2016. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, 372–387. IEEE.

Pintor, M.; Roli, F.; Brendel, W.; and Biggio, B. 2021. Fast Minimum-norm Adversarial Attacks through Adaptive Norm Constraints. *arXiv preprint arXiv:2102.12827*.

Podschwadt, R.; and Takabi, H. 2019. On Effectiveness of Adversarial Examples and Defenses for Malware Classification. In *International Conference on Security and Privacy in Communication Systems*, 380–393. Springer.

Pooladian, A.-A.; Finlay, C.; Hoheisel, T.; and Oberman, A. M. 2020. A principled approach for generating adversarial images under non-smooth dissimilarity metrics. In *AISTATS*.

Rao, S.; Stutz, D.; and Schiele, B. 2020. Adversarial Training against Location-Optimized Adversarial Patches. In *ECCV Workshop on the Dark and Bright Sides of Computer Vision: Challenges and Oppurtinities for Privacy and Security*.

Rastrigin, L. 1963. The convergence of the random search method in the extremal control of a many parameter system. *Automaton & Remote Control*, 24: 1337–1342.

Rauber, J.; Brendel, W.; and Bethge, M. 2017. Foolbox: A Python toolbox to benchmark the robustness of machine learning models. In *ICML Reliable Machine Learning in the Wild Workshop*.

Rebuffi, S.-A.; Gowal, S.; Calian, D. A.; Stimberg, F.; Wiles, O.; and Mann, T. 2021. Fixing data augmentation to improve adversarial robustness. *arXiv preprint arXiv:2103.01946*.

Rony, J.; and Ben Ayed, I. 2020. Adversarial Library.

Rony, J.; Hafemann, L. G.; Oliveira, L. S.; Ayed, I. B.; Sabourin, R.; and Granger, E. 2019. Decoupling direction and norm for efficient gradient-based l2 adversarial attacks and defenses. In *CVPR*, 4322–4330.

Schott, L.; Rauber, J.; Bethge, M.; and Brendel, W. 2019. Towards the first adversarially robust neural network model on MNIST. In *ICLR*.

Stokes, J. W.; Wang, D.; Marinescu, M.; Marino, M.; and Bussone, B. 2017. Attack and defense of dynamic analysis-based, adversarial neural malware classification models. *arXiv preprint arXiv:1712.05919*.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2014. Intriguing properties of neural networks. In *ICLR*, 2503–2511.

Thys, S.; Van Ranst, W.; and Goedemé, T. 2019. Fooling automated surveillance cameras: adversarial patches to attack person detection. In *CVPR Workshops*.

Tu, C.-C.; Ting, P.; Chen, P.-Y.; Liu, S.; Zhang, H.; Yi, J.; Hsieh, C.-J.; and Cheng, S.-M. 2019. Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. In *AAAI*.

Uesato, J.; O'Donoghue, B.; Van den Oord, A.; and Kohli, P. 2018. Adversarial risk and the dangers of evaluating against weak attacks. In *ICML*.

Wong, E.; Schmidt, F. R.; and Kolter, J. Z. 2019. Wasserstein Adversarial Examples via Projected Sinkhorn Iterations. In *ICML*.

Yang, C.; Kortylewski, A.; Xie, C.; Cao, Y.; and Yuille, A. 2020. PatchAttack: A Black-box Texture-based Attack with Reinforcement Learning. In *ECCV*.

Zabinsky, Z. B. 2010. Random search algorithms. *Wiley encyclopedia of operations research and management science*.

Zajac, M.; Zołna, K.; Rostamzadeh, N.; and Pinheiro, P. O. 2019. Adversarial framing for image and video classification. In *AAAI*, 10077–10078.

Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 586–595.

Zhao, P.; Liu, S.; Chen, P.-Y.; Hoang, N.; Xu, K.; Kailkhura, B.; and Lin, X. 2019. On the design of black-box adversarial examples by leveraging gradient-free optimization and operator splitting method. In *ICCV*, 121–130.