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 \( \times \) 20 adversarial patches and 2-pixel wide adversarial frames for 224 \( \times \) 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.

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

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

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

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

with $t$ as target class. Generating such $\delta$ can be translated into an optimization problem as

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

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 $T$.

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 $l_\infty$-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 $l_\infty$- and $l_2$-bounded black-box attacks. It uses random search to iteratively generate samples on the surface of the $l_\infty$- 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 $D$, an iteration of RS at step $i$ is given by

$$\delta \sim D(x^{(i)}), \quad x^{(i+1)} = \arg \min_{y \in \{x^{(i)}, x^{(i)} + \delta\}} L(y). \quad (2)$$

At every step an update of the current iterate $x^{(i)}$ is sampled according to $D$ 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 $\delta$ 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 $D$ 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

\[\begin{array}{l}
\text{input : loss } L, \text{ input } x_{\text{orig}}, \text{ max query } N, \text{ sparsity } k, \text{ input space constraints } S \\
\text{output: approximate minimizer of } L \\
1. M \leftarrow k \text{ indices of elements to be perturbed} \\
2. \Delta \leftarrow \text{values of the perturbation to be applied} \\
3. z \leftarrow x_{\text{orig}}, \ z_M \leftarrow \Delta \quad \text{// set elements in} \\
M \text{ to values in } \Delta \\
4. L^* \leftarrow L(z), \ i \leftarrow 0 \quad \text{// initialize loss} \\
5. \text{while } i < N \text{ and success not achieved do} \\
6. \quad M' \leftarrow \text{sampled modification of } M \\
7. \quad \Delta' \leftarrow \text{sampled modification of } \Delta \\
8. \quad z \leftarrow x_{\text{orig}}, \ z_{M'} \leftarrow \Delta' \\
9. \quad \text{// create new candidate in } S \\
10. \quad \text{if } L(z) < L^* \text{ then} \\
11. \quad \quad L^* \leftarrow L(z), \ M \leftarrow M', \ \Delta \leftarrow \Delta' \quad \text{// if} \\
12. \quad \quad \text{loss improves, update sets} \\
13. \quad i \leftarrow i + 1 \\
14. \text{return } z
\end{array}\]

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 \(\Delta\) 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 \(\Delta\) (step 7). In some threat models (e.g. adversarial frames) the set \(M\) cannot be changed, so \(M' \equiv M\) at every step. How \(\Delta'\) 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 \(S\)). This follows the intuition that larger perturbations should lead faster to an adversarial example. Moreover, the difference of the candidates \(M'\) and \(\Delta'\) with \(M\) and \(\Delta\) 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 targeted attacks the margin loss \(L_{\text{margin}}(f(\cdot), y) = f_y(\cdot) - \max_{x \neq y} f(\cdot), \) where \(y\) is the correct class, so that \(L < 0\) is equivalent to misclassification, whereas for targeted attacks we use the cross-entropy loss \(L_{\text{CE}}\) 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 \(\Delta \sim U([0, 1]^{l_0})\), 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 \(B \subset U \setminus A\), with \(|A| = |B| = \alpha^{(i)} \cdot k\), and create \(M' = (M \setminus A) \cup B\). \(\Delta'\) 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 \(\alpha_{\text{init}}\), 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 \(\alpha_{\text{init}}\) 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 “gray-scale” 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 \(224 \times 224\), 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 \(8 \times \#\text{pixels}\) queries only for the
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.

**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), 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 which is only 0.1% of the total number of pixels.

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.

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.

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

**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 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.

![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).](image)

![Figure 3: Untargeted \( l_0 \)-adversarial examples generated by our pixel-based \( l_0 \)-RS algorithm for \( k \) = 50 pixels.](image)
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 $l_\infty$ 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.

### 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.

<table>
<thead>
<tr>
<th>attack</th>
<th>type</th>
<th>$l_0$-AT ResNet</th>
<th>$l_1$-AT ResNet</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>$l_0$-bound in pixel space $k = 24$</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PGD</td>
<td>white-box</td>
<td>68.7%</td>
<td>72.7%</td>
</tr>
<tr>
<td>CornerSearch</td>
<td>black-box</td>
<td>59.3%</td>
<td>64.9%</td>
</tr>
<tr>
<td>$l_0$-RS</td>
<td>black-box</td>
<td>85.7%</td>
<td>81.0%</td>
</tr>
<tr>
<td><strong>$l_0$-bound in feature space $k = 24$</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VFGA</td>
<td>white-box</td>
<td>40.5%</td>
<td>27.5%</td>
</tr>
<tr>
<td>FMN</td>
<td>white-box</td>
<td>52.9%</td>
<td>28.2%</td>
</tr>
<tr>
<td>FDPGD</td>
<td>white-box</td>
<td>46.4%</td>
<td>26.9%</td>
</tr>
<tr>
<td>CornerSearch</td>
<td>black-box</td>
<td>43.2%</td>
<td>29.4%</td>
</tr>
<tr>
<td>$l_0$-RS</td>
<td>black-box</td>
<td>63.4%</td>
<td>38.3%</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>attack</th>
<th>type</th>
<th>$k = 12$ (pixels)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pointwise Attack</td>
<td>black-box</td>
<td>ABS 31%</td>
</tr>
<tr>
<td>CornerSearch</td>
<td>black-box</td>
<td>ABS 29%</td>
</tr>
<tr>
<td>$l_0$-RS</td>
<td>black-box</td>
<td>ABS 55%</td>
</tr>
</tbody>
</table>

### 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:

$$\text{arg min } y \langle w_x + \delta, x \rangle = \text{arg min } \langle yw_x, \delta \rangle$$

$$\|\delta\|_{l_0} \leq k$$

where $\delta \in \{0, 1\}$.

$$\text{arg min } \langle yw_x, \delta \rangle, \quad \|\delta\|_{l_0} \leq k$$

$$\delta \in \{0, 1\}$$

$$w_x$$

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 $\tilde{w}_x$. The black-box case, where $w_x$ is unknown and we are only allowed to query the model predictions $\langle \tilde{w}_x, z \rangle$ for any $z \in \mathbb{R}^d$, is more complicated since the naive weight estimation algorithm requires $O(d)$ queries to first estimate $\tilde{w}_x$ 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:

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

the location of the patch for each image independently.

and Schiele 2020) where one optimizes both the content and
image-specific patches as in (Yang et al. 2020; Rao, Stutz,
which we consider in the next section, we focus first on
less of the image and the position where they are applied,
aim at universal patches, which fool the classifier regard-

(Brown et al. 2017; Karmon, Zoran, and Goldberg 2018)
but can be changed in an arbitrary way. While some works
square-shaped, and limited to a small portion of the image
there the perturbed pixels are localized, often

Another type of sparse attacks which recently received
attentions are adversarial patches introduced by (Brown
et al. 2017). There the perturbed pixels are localized, often

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% - - 82.7% - -

attacks, they have been optimized for rather small perturba-
tions; thus, they can be plugged in there. We use Square
Attack (SA) (An-
driushchenko 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 + SA 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 \( l_{\infty} \)-attacks, they have been optimized for rather small perturba-
tions whereas for patches all pixels can be manipulated arbi-
trarily in \([0, 1] \). Thus in principle any black-box method for an \( l_{\infty} \)-threat model can be plugged in there. We use Square
Attack (SA) (An-
driushchenko 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 + SA 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 \( l_{\infty} \)-attacks, they have been optimized for rather small perturba-
tions whereas for patches all pixels can be manipulated arbi-
trarily in \([0, 1] \). Thus in principle any black-box method for an \( l_{\infty} \)-threat model can be plugged in there. We use Square
Attack (SA) (An-
driushchenko 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 + SA 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 \( l_{\infty} \)-attacks, they have been optimized for rather small perturba-
tions whereas for patches all pixels can be manipulated arbi-
trarily in \([0, 1] \). Thus in principle any black-box method for an \( l_{\infty} \)-threat model can be plugged in there. We use Square
Attack (SA) (An-
driushchenko 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 + SA 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 \( l_{\infty} \)-attacks, they have been optimized for rather small perturba-
tions whereas for patches all pixels can be manipulated arbi-
trarily in \([0, 1] \). Thus in principle any black-box method for an \( l_{\infty} \)-threat model can be plugged in there. We use Square
Attack (SA) (An-
driushchenko 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 + SA 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 \( l_{\infty} \)-attacks, they have been optimized for rather small perturba-
tions whereas for patches all pixels can be manipulated arbi-
trarily in \([0, 1] \). Thus in principle any black-box method for an \( l_{\infty} \)-threat model can be plugged in there. We use Square
Attack (SA) (An-
driushchenko 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 + SA 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 \( l_{\infty} \)-attacks, they have been optimized for rather small perturba-
tions whereas for patches all pixels can be manipulated arbi-
trarily in \([0, 1] \). Thus in principle any black-box method for an \( l_{\infty} \)-threat model can be plugged in there. We use Square
Attack (SA) (An-
driushchenko 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 + SA 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.
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 $20 \times 20$ and query limit of 10,000 and for targeted attacks (random target class for each image) with patch size $40 \times 40$ 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 $50 \times 50$ 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.

<table>
<thead>
<tr>
<th>Attack</th>
<th>VGG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transfer PGD</td>
<td>3.3%</td>
</tr>
<tr>
<td>Transfer MI-FGSM</td>
<td>1.3%</td>
</tr>
<tr>
<td>PGD w/ GE</td>
<td>35.1%</td>
</tr>
<tr>
<td>ZO-AdaMM</td>
<td>45.8%</td>
</tr>
<tr>
<td>Sparse-RS + SH</td>
<td>63.9%</td>
</tr>
<tr>
<td>Sparse-RS + SA</td>
<td>72.9% ± 3.6</td>
</tr>
<tr>
<td>Patch-RS</td>
<td>70.8% ± 1.3</td>
</tr>
</tbody>
</table>

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


