How Many Perturbations Break This Model? Evaluating Robustness Beyond Adversarial Accuracy

Raphael Olivier [1] **Bhiksha Raj** [1]

Abstract

Robustness to adversarial attacks is typically evaluated with adversarial accuracy. While essential, this metric does not capture all aspects of robustness and in particular leaves out the question of how many perturbations can be found for each point. In this work, we introduce an alternative approach, adversarial sparsity, which quantifies how difficult it is to find a successful perturbation given both an input point and a constraint on the direction of the perturbation. We show that sparsity provides valuable insight into neural networks in multiple ways: for instance, it illustrates important differences between current state-of-the-art robust models them that accuracy analysis does not, and suggests approaches for improving their robustness. When applying "broken" defenses effective against weak attacks but not strong ones, sparsity can discriminate between the "totally ineffective" and the "partially effective" defenses. Finally, with sparsity we can measure increases in robustness that do not affect accuracy: we show for example that data augmentation can by itself increase adversarial robustness, without using adversarial training.

1. Introduction

[1]Designing adversarially robust machine learning models has become one of the main objectives of the research community. Adversarial examples, these slightly perturbed inputs that pose significant problems for output prediction, are the source of multiple security threats: not only are they dangerous in and of themselves, but they also contribute to enabling data poisoning attacks (Shafahi et al., 2018) and even membership inference attacks (Choquette-Choo et al., 2021). Making models robust to adversarial perturbations is very difficult for multiple reasons; one of them is that the proper evaluation method of robustness is not a trivial problem.

A few years ago, it was common to evaluate defenses against a variety of adversarial attacks, like FGSM ((Goodfellow et al., 2014)), DeepFool ((Moosavi-Dezfooli et al., 2016)) or JSMA ((Papernot et al., 2016)). Some defenses could claim good results on some attacks, but remained vulnerable to others (e.g. Papernot et al. (2015) broken in Carlini & Wagner (2016a)), suggesting multiple aspects to robustness and giving adversarial research the form of an "arms race" between attacks and defenses. Nowadays however, this approach has lost popularity and it is considered good practice to focus on the accepted strongest attacks for evaluation: typically PGD ((Madry et al., 2018)) or its step-size free variant APGD (Croce & Hein, 2020) for bounded attacks and Carlini&Wagner ((Carlini & Wagner, 2016b)) for unbounded attacks.

We argue that this approach ignores important aspects of adversarial robustness. PGD for instance is considered a surrogate to the worst-case accuracy given a fixed threat model. With input x, label y and a set of admissible perturbations Δ, worst case accuracy on x is equal to 1 if:

$$\forall \delta \in \Delta,\ f(x+\delta) = y \tag{1}$$

and 0 otherwise. This metric reflects whether, in the vicinity of an input point, there is *one* successful perturbation. This can be misleading in the quest for robustness. Consider a hypothetical defense that, around every point, eliminates 99% of all dangerous perturbations, leaving 1% to find. Its worst-case accuracy would be 0%, just like an undefended model, and the defense would be considered "broken" by the research community. In other words, worst-case accuracy is biased towards defenses that are totally effective for some points, and against those that are partially effective around all points. Whether a point has any or no adversarial perturbation is relevant knowledge, but it leaves out a lot of information. The "broken" defense above may for instance be useful in real-world contexts, where perturba-

[1]Language Technologies Institute, Carnegie Mellon University, Pittsburgh, USA. Correspondence to: Raphael Olivier <rolivier@cs.cmu.edu>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

[1]Our code is available at https://github.com/RaphaelOlivier/sparsity

tions are harder to craft than on research datasets. Moreover, it may hypothetically be complementary to other "broken" defenses that eliminate a distinct subset of perturbations or improve a defense with non-zero adversarial accuracy.

In this work, we propose an alternative approach for measuring adversarial robustness. Rather than measuring whether around an input x there is *at least one* adversarial perturbation, we try to estimate *how many* such perturbations there are, i.e. the *size* of the set of successful adversarial perturbations. In high-dimensional input spaces, there are not many informative and computationally tractable metrics to measure this size. Traditional measure theory hits considerable difficulties: for instance, within an L_2-ball of radius ϵ around x, the set of adversarial examples is way too small to be estimated with rejection sampling.

A naive approach would be to revert to evaluating with a set of more or less strong attacks and claim for instance that a model robust to FGSM but not PGD likely has fewer adversarial perturbations than a model robust to neither. However, there are good reasons why this approach was abandoned. Many defenses implicitly rely on gradient obfuscation effects (Athalye et al., 2018) which makes attack convergence artificially difficult without actually defending against perturbations. Using only strong attacks, possibly enhanced with adaptive methods, protects evaluation against such effects.

In our approach, rather than weakening the attack we propose to constrain it to only look at a subset of the set of admissible perturbations Δ. The question we try to answer is: how large a typical subset must be to find an adversarial perturbation in it? We define metrics quantifying the expected size of the subset: the bigger it is, the fewer perturbations there are. We call this metric *adversarial sparsity*.

A major challenge is to define a distribution of subsets and a metric adapted to the adversarial threat model. In section 3, we define sparsity formally for L_2 and L_∞ perturbations. In the L_2 case we consider uniform directions u on the L_2 sphere, and adversarial sparsity is the expected angle $\widehat{\langle \delta, u \rangle}$ needed to find a successful perturbation. For L_∞ attacks we consider vertices of the L_∞ cube $u \in \{\pm 1\}^n$, and measure the number of dimensions k to modify to find a successful perturbation. We also discuss in section 3 how adversarial sparsity indeed reflects the size of the successful perturbations set.

Like worst-case accuracy, sparsity is not directly tractable, so we must estimate it using modified PGD attacks on a random sample of directions. Our algorithm is detailed in section 4. Armed with this tool we revisit multiple models and defenses proposed in prior works on the CIFAR10 dataset (section 5). We both include very strong defenses, taken among the strongest models on the RobustBench leaderboard (Croce et al., 2021); much weaker defenses that are considered "broken"; and undefended models trained with various data augmentation schemes. Using sparsity we discover or strengthen several properties of these models (section 6), among which:

- Higher adversarial accuracy does *not* necessarily mean much fewer adversarial perturbations: many improvements on robustness benchmarks consist of removing a few residual perturbations around "almost robust" inputs.
- Most of the defenses "broken" by strong attacks do not reduce *at all* the number of perturbations, indicating that robustness claims on these defenses are spurious.
- Data augmentation methods, known to improve the results of adversarial training (Rebuffi et al., 2021), in fact increase robustness even *without* adversarial training - though not enough to be reflected in adversarial accuracy.

These findings could lead to promising developments in future robust models. They confirm the interest in alternative evaluation metrics for adversarial robustness.

2. Related Work

2.1. Weak and strong attacks and defenses

Multiple adversarial attacks have been proposed since the phenomenon was popularized by Szegedy et al. (2014). Early works include for instance the Fast Gradient-Sign Method (FGSM) attack (Goodfellow et al., 2014). However, the two standard attacks for white-box defense evaluation in recent years are Projected Gradient Descent (PGD) (Madry et al., 2018) for norm-bounded attacks, and Carlini&Wagner (CW) (Carlini & Wagner, 2016b) for regularization-based unbounded attacks.

As it turns out, most defenses can be broken using these attacks, possibly in an adaptive fashion (Athalye et al., 2018). One recent improvement in attack success was provided by AutoAttack (Croce & Hein, 2020), an ensemble of attacks mostly based on PGD, which adaptively changes hyperparameters during optimization. Other recent attacks typically bring orthogonal improvement, such as perception-aligned attacks (Göpfert et al., 2020).

Early work on adversarial attacks and defenses took the form of an arms race. Defenses breaking current state-of-the-art attacks were proposed, like adversarial training (Goodfellow et al., 2014), input transformations (Xie et al., 2017; Guo et al., 2017) or projection to the data manifold (Samangouei et al., 2018) only to be broken soon after with a new attack

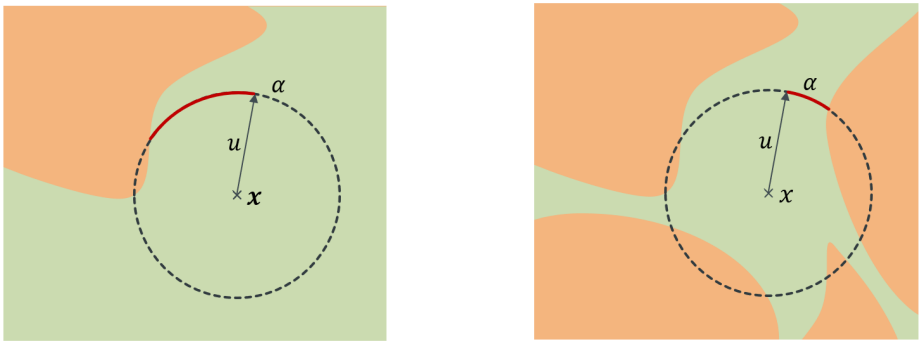


Figure 1. Representation of the decision boundary of some binary classification task (regions in green and orange) around point x in two cases. In both cases, the model is not robust around x with the represented L_2 radius ϵ. However, there are many more adversarial perturbations on the right than on the left. Adversarial accuracy evaluates both cases at 0, but α, the angle between fixed direction u and the closest adversarial perturbation, is smaller on the right. Adversarial sparsity is the expected value of α when sampling u uniformly.

or method (Athalye et al., 2018). For reasons mentioned above, we focus on the strong PGD attack in this work.

More recent works follow a somehow different trend. On the one hand, numerous proposed defenses are broken with attacks that already exist, for lack of rigorous evaluation. On the other hand, many works focus on promising approaches that have shown to be empirically or sometimes provably robust. Such approaches include randomized smoothing (Cohen et al., 2019), exact or relaxed certified methods (Kolter & Wong, 2017), and PGD-based adversarial training (Madry et al., 2018; Wong et al., 2020) which can be enhanced with data augmentation (Gowal et al., 2020; Rebuffi et al., 2021). In this work, we explore both adversarial training and older, supposedly broken defenses, and revisit the extent of this non-robustness.

2.2. Geometry of adversarial examples

The idea to explore the spherical geometry of adversarial examples has some precedents in the literature. Tramèr et al. (2017) take a linear algebra approach and estimate the dimension of a subspace of adversarial examples around a perturbation. Khoury & Hadfield-Menell (2018), and to an extent Samangouei et al. (2018) explore the data manifold and attempt to explain adversarial examples as out-of-manifold points that models have trouble classifying. Such approaches that consider adversarial perturbations as artifacts have been challenged by works such as Ilyas et al. (2019) which demonstrate that adversarial perturbations are features that models can learn, and as such are reasonable input points.

We do not make claims about the "nature" of adversarial examples, but simply propose metrics to quantify them. This makes this work closer to Tramèr et al. (2017) in its principle, although sparsity is different than dimension estimation. Parallels may also be drawn with Deep Hypersphere Embeddings (Liu et al., 2017), which have been shown to have some robustness properties (Pang et al., 2020).

2.3. Alternatives to adversarial accuracy

The limitations of adversarial accuracy have led several previous works to propose alternative robustness metrics. We discuss those metrics and how they differ from ours.

Minimal perturbation: it is common to compute the closest adversarial example to a given input as a metric for robustness around this point. Some attack algorithms are based on that approach (Carlini & Wagner, 2016b). This notion differs from sparsity in that it removes the notion of attack bound and perturbation constraint altogether: instead, all points are potential adversarial examples and the attacker finds the "best" one. A model could have a very small minimal perturbation but still have high sparsity for large radii.

Probabilistic robustness: Robey et al. (2022) introduce a relaxation on "worst-case" robustness, by training models to be robust to most perturbations sampled uniformly, rather than all of them. Stemming from similar motivations to our work, probabilistic robustness differs from sparsity significantly in its approach, as it is "non-adversarial" robustness. Where we sample geometric regions to constrain the adversary, probabilistic robustness samples perturbations directly. Indeed, Robey et al. (2022) show that probabilistic robustness and adversarial robustness are largely uncorrelated, contrary to sparsity and accuracy.

Local Intrinsic Dimensionality (LID): Ma et al. (2018) analyze the local dimension of the submanifold of adversarial perturbations around an input. They use this metric to better understand the properties of adversarial regions. sparsity differs from LID as it does not assume that adversarial

examples lie on a manifold. We probe regions and try to find adversarial examples in them, while LID starts from an adversarial example and probes the "adversarial region" around it. These approaches are intuitively complementary, in that sparsity estimates "how many" adversarial regions there are and LID "how big" those regions are. This duality is an interesting research topic for future work.

3. Adversarial Sparsity

3.1. Definitions

Throughout the following sections f designates a machine learning model, and L its loss function. ϵ is the radius of the attack, and Δ is the set of admissible perturbations. Usually $\Delta = B_k^n$ with B_k^n the n-dimensional hyperball in norm L_k:

$$B_\infty^n = [0,1]^n$$

$$B_2^n = \{x \in \mathbb{R}^n \mid x_1^2 + ... + x_n^2 \leq 1\}$$

where x_j is the i^{th} coordinate of x.

To make sparsity computations simpler, we will instead only consider the adversarial examples that use their entire perturbation budget. This means that for $k < \infty$ $\Delta = S_k^n$, using the hypersphere ($\|x\| = 1$) rather than the hyperball ($\|x\| \leq 1$). For $k = \infty$ $\Delta = \{\pm 1\}^n$, i.e. we consider only the (finite) set of vertices of the hypercube. This is done with little cost of generality, as strong attacks like PGD nearly always use all of their perturbation budget (and the simpler FGSM attack is constrained to do so).

Given a point x and a radius ϵ, we define the *adversarial set* as the set of successful admissible perturbations:

$$\text{Adv}(f, x, \epsilon) := \{\delta \in \Delta \mid f(x + \epsilon.\delta) \neq f(x)\} \quad (2)$$

In our convention, adversarial perturbations are always unit vectors, to be scaled by factor ϵ when applied. Finally, we can redefine adversarial accuracy over a point x within this formalism:

$$\text{AA}(f, x, \epsilon) := \mathbf{1}[\text{Adv}(f, x, \epsilon) = \varnothing] \quad (3)$$

We will also sample points from probability distributions. $\mathcal{U}(X)$ designates the uniform distribution over measurable set X.

3.2. Defining sparsity

Adversarial sparsity quantifies "how big" a subset of Δ typically needs to be to contain an adversarial perturbation. Formally, we assume access to a sequence of increasing subsets of Δ: with $m_1 < m_2$, $\emptyset \subseteq \Delta^{m_1} \subseteq \Delta^{m_2} \subseteq \Delta$. We can define adversarial sparsity relative to Δ^m as:

$$\text{AS}(f, x, \epsilon, (\Delta^m)) := \inf\{m \mid \Delta^m \cap \text{Adv}(f, x, \epsilon) \neq \emptyset\} \quad (4)$$

If we have a distribution $\mathcal{D}$ of such sequences, we can define adversarial sparsity as the expected value of AS

$$\overline{\text{AS}}(f, x, \epsilon) := \mathbb{E}_{(\Delta^m) \sim \mathcal{D}}[\text{AS}(f, x, \epsilon, (\Delta^m))] \quad (5)$$

Intuitively, larger sparsity means more robust models, provided that some reasonable constraints are enforced on distribution $\mathcal{D}$. In particular, the distribution should be isotropic, i.e. not favor any particular direction.

We now give more concrete definitions relative to our two threat models. For L_2 perturbations we consider angular constraints. We sample $u \sim \mathcal{U}(S_2^n)$ and for $\alpha \in [0, \pi]$ we define Δ^α as the spherical cap of direction u and angle α, that is the set of admissible perturbations that form an angle at most α with u:

$$\Delta^\alpha = \text{Sc}(u, \alpha) := \{\delta \in S_2^n \mid \delta \cdot u \geq \cos\alpha\} \quad (6)$$

For L_∞ perturbations, we sample both a vertex $u \in \mathcal{U}(\{\pm 1\}^n)$ and a permutation of dimensions $\sigma \in \mathfrak{S}_n$. For $m \in \{1, ..., n\}$ we define Δ^m as the set of perturbations differing of u only over dimensions $\sigma(1), ..., \sigma(m)$:

$$\Delta^m := \{\delta \in S_\infty^n \mid \forall k > m \ \delta_{\sigma(k)} = u_{\sigma(k)}\} \quad (7)$$

In other words Δ^m is the set of perturbations differing from u by at most m specific pixels $\sigma(1), ..., \sigma(m)$. The permutation σ is required to enforce that all dimensions are equally probable under the distribution.

3.3. Sparsity and number of perturbations

We propose sparsity as an approach to measure the size of the adversarial set. Is there indeed a relationship between the two? Intuitively this depends on how that set itself is distributed over Δ. If for instance $\text{Adv}(f, x, \epsilon)$ is "one half" of Δ (e.g. all perturbations where the bottom left pixel perturbation is positive) then its sparsity will be 0 for half the directions but quite large for many others, leading to an expected sparsity that does not accurately reflect the immense size of this adversarial set. So which conditions should be met for sparsity to be a relevant metric?

Let us consider a simplified case where the set of perturbations is finite: $\text{Adv}(f, x, \epsilon) = \{\delta_1, ..., \delta_k\}$. If we assume that the δ_i are uniformly sampled over the admissible set, then there are no preferred directions of high adversarial concentration, as opposed to the example above. In this case, the link between sparsity and k can be mathematically established. For L_2 perturbations, the expected value of $\text{AS}(f, x, \epsilon)$ when sampling δ_j is:

$$\mathbb{E}[\text{AS}(f, x, \epsilon)] = \int_0^{\frac{\pi}{2}} ((1 - t_\alpha)^k + (t_\alpha)^k) d\alpha \quad (8)$$

with $t_\alpha = I_{sin^2(\alpha)}(\frac{n-1}{2}, \frac{1}{2})$ where I is the incomplete regularized beta function. Meanwhile for L_∞ perturbations:

$$\mathbb{E}[\text{AS}(f, x, \epsilon)] = \sum_{m=0}^{n} (1 - 2^{-m})^k \tag{9}$$

In this case, we can also show that :

$$\frac{n - \log_2 k}{4} \leq \mathbb{E}[\text{AS}(f, x, \epsilon)] \leq n - \log_2 k + \frac{e}{e-1} \tag{10}$$

i.e. sparsity tends to vary like $n - \log_2 k$. We defer all proofs to appendix E.

In the general case $\text{Adv}(f, x, \epsilon)$ is infinite[2] and we are interested in its volume relative to Δ, rather than its cardinal. Both situations can be linked if considering for instance

$$\text{Adv}(f, x, \epsilon) = \bigcup_{j=1}^{k} \text{Sc}(\delta_j, \beta) \tag{11}$$

ie $\text{Adv}(f, x, \epsilon)$ is the union of spherical caps of equal radius β. β can be called the "local adversarial radius", i.e. how much a typical adversarial perturbation should be changed to recover the original input. In that case, the volume of $\text{Adv}(f, x, \epsilon)$ is equal to $k.V(\text{Sc}(u, \beta))$ (assuming disjoint union) and $\text{AS}(f, x, \epsilon)$ would vary by at most β compared to the finite case. In other words, sparsity depends on both k and β. Therefore comparing the sparsity of two models is akin to comparing the size of their adversarial sets provided that these sets have similar local radii. We implicitly make that assumption in our experiments: estimating the local adversarial radius as well as sparsity would be an interesting extension of this work.

4. Algorithms

In this section, we detail the algorithms we use to empirically compute sparsity. We first describe the Projected Gradient Descent (PGD) attack, and the modifications we implement to compute adversarial examples in constrained regions. Then we describe the full sparsity computation method, which applies PGD with multiple constraints. We sum up the full procedure in L_2 norm in Algorithm 1.

4.1. Projected Gradient Descent

The PGD attack (Madry et al., 2018) is a strong first-order attack, i.e. it only uses model gradient information. It optimizes the following non-convex objective:

$$\arg\max_{\|\delta\| < \epsilon} L(f(x + \delta), y) \tag{12}$$

[2]With our definition of the admissible set, $\text{Adv}(f, x, \epsilon)$ is finite in the L_∞ case. But the same reasoning applies if we consider the more general set $\Delta = S_\infty^n$

using projected gradient descent for a number n of gradient update steps. At each step $k+1$ consists of a gradient optimization step followed by a projection step which depends on the attack norm:

$$\delta'_k \leftarrow \delta_k + \eta.\text{sign}(\nabla_{\delta_k} L(f(x + \delta_k), y)) \ \text{(grad. ascent)} \tag{13}$$

$$\delta_{k+1} \leftarrow \text{clip}(\delta'_k, \epsilon) \ (L_\infty \text{ projection}) \tag{14}$$

$$\delta_{k+1} \leftarrow \min(1, \frac{\epsilon}{\|\delta'_k\|_2}).\delta'_k \ (L_2 \text{ projection}) \tag{15}$$

4.2. Constrained PGD

To practically compute angular sparsity we need to design a constrained attack, able to project not on a hyperball but on the subsets defined in section 3.2.

For L_∞ attacks this is straightforward: given a vertex u, a perturbation σ, and a number of dimensions m, we append after projection an additional step enforcing the constraints:

$$\forall k > m \ \ \delta_{\sigma(k)} \leftarrow u_{\sigma(k)} \tag{16}$$

For L_2 attacks, the steps are slightly more complex, as we need an angular projection on the spherical cap of angle α and direction u. This requires replacing the projection step in equation 15 with a projection by both norm and angle. The technical steps of that projection are detailed in Appendix A

We name PGD_m and PGD_α these constrained attacks in the following.

4.3. Computing sparsity over a point

To estimate sparsity for L_2 (resp. L_∞) attacks, given a direction u (resp. u, σ) we explore possible values of α (resp. m) with binary search, and at each step run the constrained attack. We average over multiple sampled directions to estimate $\overline{\text{AS}}$ (in general 100).

The equivalent L_∞ algorithm is similar, replacing only direction u by (u, σ), angle α by a number of pixels m and the constrained L_2-PGD by the constrained L_∞ PGD. The time complexity of these algorithms is discussed in Appendix B.

4.4. Computing sparsity over a dataset

Adversarial sparsity only makes sense if there *are* adversarial perturbations in the vicinity of an input x. It is designed to capture the level of vulnerability in non-robust models. We choose to restrict sparsity computation to the residual subset of inputs where the model is vulnerable. For instance, if a model is robust over 50 inputs and has sparsity 0.3 over another 50, we will say it reaches 50% accuracy and *residual*

Algorithm 1 L_2 sparsity computation algorithm

Require: model f, point (x, y), $K \in \mathbb{N}, N \in \mathbb{N}$
  $k \leftarrow 0$
  $i \leftarrow 0$
  while $i \leq N$ **do**
    $\alpha_0^i \leftarrow 0, \alpha_1^i \leftarrow \pi, u_i \sim \mathcal{U}(S^n)$
    $i \leftarrow i + 1$
  end while
  while $k \leq K$ **do**
    $i \leftarrow 0$
    while $i \leq N$ **do**
      $\alpha^i \leftarrow \frac{\alpha_0^i + \alpha_1^i}{2}$
      $x_{\text{adv}} \leftarrow \text{PGD}_{\alpha^i}(f, x, y, u_i)$
      if $f(x_{\text{adv}}) \neq y$ **then**
        $\alpha_1^i \leftarrow \alpha^i$
      else
        $\alpha_0^i \leftarrow \alpha^i$
      end if
      $i \leftarrow i + 1$
    end while
    $k \leftarrow k + 1$
  end while
  return $\frac{1}{n} \sum_0^n \alpha^i$

sparsity 0.3. An alternative could be to default sparsity to a max value when evaluating a model that is robust around x ($\text{Adv}(f, x, \epsilon) = \varnothing$): π for L_2 attacks, and dimension n for L_∞ attacks.

5. Experiments

We run experiments in L_2 and L_∞ perturbations on the CIFAR10 dataset (Krizhevsky et al., 2009). Additional results on ImageNet can be found in Appendix C. We use attack radii $\epsilon = 0.5$ and $\epsilon = 8/255$ respectively.

5.1. Models and defenses

We mainly evaluate defenses over a classic residual CNN architecture: ResNet-18 (He et al., 2016). We evaluate larger ResNets and WideResNets as well. We consider multiple defense mechanisms:

5.1.1. Adversarial training

Adversarial training consists in applying adversarially perturbed inputs during training rather than or along with natural inputs. It is one of the most robust defenses against PGD attacks. We use multiple pretrained models using state-of-the-art defenses based on adversarial training (Gowal et al., 2020; Augustin et al., 2020; Rebuffi et al., 2021; Rade & Moosavi, 2021; Huang et al., 2021). These defenses are well ranked in the Robustbench leaderboard (Croce et al., 2021) for L_2 attacks on CIFAR10. Some use additional extra data for pretraining. We also train ResNet-18 models with PGD training, following (Madry et al., 2018), with 1 or 10 attack steps.

5.1.2. Preprocessing

Early defenses often used input preprocessing during testing to "erase" adversarial noise. These defenses have mostly been beaten by stronger or adaptive attacks. We revisit some of them: JPEG compression and decompression (Guo et al., 2017), Feature Squeezing, and Spatial Smoothing (Xu et al., 2018). The latter two were proposed as complementary defenses. This makes analysis with sparsity particularly interesting to determine whether each defense has a partial effect on robustness and would benefit from ensembling.

5.1.3. Data augmentation

(Xie et al., 2017) suggest that random transformations can mitigate adversarial attacks. This defense is also considered broken when using strong attacks (Athalye et al., 2018). Additionally, (Rebuffi et al., 2021) showed that data augmentation can significantly improve the performance of adversarial training.

We wonder if data augmentation affects robustness by itself. We train multiple models with standard training and data augmentation schemes used in (Xie et al., 2017) and (Rebuffi et al., 2021), and evaluate them with sparsity.

5.2. Attack

Preprocessing methods (JPEG, Feature Squeezing, Spatial Smoothing) may pose obfuscation problems when computing adversarial attacks during sparsity estimation. Hence, we follow (Shin & Song, 2017) and use Differentiable JPEG, which we backpropagate through. We use the Straight-Through estimator (Athalye et al., 2018) against the other two.

In addition to sparsity, we evaluate adversarially trained models using the AutoAttack (Croce & Hein, 2020) and $\epsilon = 0.5$ or $\epsilon = 8/255$. To reduce computation time we run a slightly cheaper AutoAttack, using only APGD-CE and APGD-DLR. All other models (weakly defended or undefended) achieve 0% adversarial accuracy, which we do not report in result tables.

6. Results

The results of adversarially defended models are reported in Table 1. We report averaged values of sparsity over the first 1000 vulnerable inputs in the CIFAR10 test set and, for each input, 100 random directions.

1	Model	Defenses	Clean acc.	Adv. acc.	Sparsity	Clean acc.	Adv. acc.	Sparsity
2	R-18	None	88	0	0.180	88	0	56.8
3	R-18	Madry et al. (2018) (1 step)	90	62.1	0.559	90.73	40.08	229
4	R-18	Madry et al. (2018)	88.8	67.3	0.553	80.97	47.19	277
5	R-18	Madry et al. (2018) (w/o aug.)	82.8	60.2	0.532	73.96	43.02	273
6	WR-70-16	Rebuffi et al. (2021) (w/ data)	95.74	82.3	0.581	94.16	62.91	213
7	WR-70-16	Gowal et al. (2020) (w/ data)	94.74	80.5	0.590	91.17	62.5	215
8	WR-70-16	Rebuffi et al. (2021)	92.41	80.4	0.545	89.16	65.83	202
9	WR-28-10	Rebuffi et al. (2021)	91.79	78.8	0.529	89.58	61.66	209
10	R-18	Sehwag et al. (2021)	89.5	73.4	0.539	87.08	52.08	231
11	R-18	Rade & Moosavi (2021)	90.5	76.2	0.515	92.08	57.5	209
12	WR-34-10	Augustin et al. (2020)	92.23	76.3	0.525	-	-	-
13	R-50	Augustin et al. (2020)	91.08	72.9	0.572	-	-	-
14	WR-34-R	Huang et al. (2021)	-	-	-	89.58	57.5	227
15	WR-34-R	Huang et al. (2021) (EMA)	-	-	-	89.17	57.5	230

Table 1. Evaluation of state-of-the-art adversarially trained models under L_2 ($\epsilon = 0.5$) and L_∞ ($\epsilon = 0.03$) perturbations. We report Natural accuracy (without attack), adversarial accuracy (under AutoPGD), and adversarial (residual) sparsity. Model architectures are either ResNet (R) or WideResNet (WR) . (w/o aug.) means the model was trained without any data augmentation: most models are trained with at least Crop+Resize augmentation. (w/ data) means that external data was used to train the model. For some defenses there is only a L_∞ or only a L_2-trained model available; for many both exist, in which case we report the results of both.

Defense	Test Accuracy	L_2 Sparsity	L_∞ Sparsity
None	88	0.180	56.8
JPEG	77.1	0.161	80.9
FS	87.8	0.178	60.1
SPS	74	0.147	77.7
FS+SPS	74.5	0.174	86.6

Table 2. Evaluation of ResNet18 with various "broken" defenses under L_2 attack with 20 iterations and $\epsilon = 0.5$. We report Natural accuracy (without attack) and angular sparsity for L_2 ($\epsilon = 0.5$) and L_∞ ($\epsilon = 0.03$) perturbations.

Augmentation	Accuracy	L_2 Sparsity	L_∞ Sparsity
None	88.0	0.180	56.8
Crop+Resize	93.7	0.225	64.0
Cutmix	94.58	0.185	50.8
50% Cutmix	95.83	0.202	57.0
Mixup	93.75	0.157	71.1
50% Mixup	92.5	0.202	69.4
Cutout	94.16	0.225	62.2
50% Cutout	93.75	0.225	62.6
Ricap	92.2	0.272	105
50% Ricap	91.67	0.253	83.6

Table 3. Evaluation of ResNet18 with data augmentation schemes. The augmentation is applied either on all inputs or 50% of inputs

6.1. Margin of error when estimating sparsity

Angular sparsity $\overline{\text{AS}}$ cannot be directly computed but must be approximated with the sample mean estimator over multiple directions. To assess the margin of error in this estimation we compute for multiple models, the standard deviation of L_2 sparsity over 100 directions, for 1000 input points. We find that these deviations are consistently lower than 0.022. Using the general formula for margins of errors $z * \frac{\sigma}{\sqrt{n}}$, we conclude that our estimates with 100 directions provide estimates within a ± 0.002 margin with 95% confidence. In Appendix C.3 we provide additional information on the sparsity variance, this time with respect to input points.

6.2. Comparing strong defenses by sparsity

At first sight, we can already observe in Table 1 that sparsity is overall consistent with adversarial accuracy. Adversarially defended models, whose accuracy is above 40%, have much higher sparsity than the undefended baseline. Models trained with strong adversarial training (lines 3-15) all reach L_2 (resp L_∞) sparsity greater than 0.5 (resp 200). In comparison, the baseline model achieves 0.180 (resp. 56.8).

However, among the robust models (lines 3-15), sparsity and accuracy variations are not well correlated. In fact, when it comes to L_∞ perturbations, the model with higher accuracy has often *lower* sparsity on the residual subset! We illustrate this phenomenon by plotting for robust models sparsity as a function of accuracy in Figure 2. An explanation would be that improvements in adversarial training from a baseline model A focus on robustly classifying the easiest points, i.e. those for which few perturbations fool A. Those points have higher sparsity - thus classifying them well drops the residual sparsity for the remaining points.

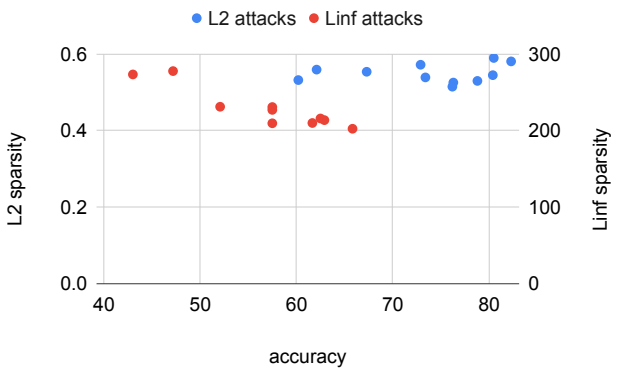


Figure 2. Sparsity as a function of adversarial accuracy for all robust models, in both L_2 norm (blue) and L_∞ norm (red). The values used are the same as in Table 1

6.3. Comparing broken defenses by sparsity

In Table 2 we compare the sparsity of the vanilla model and models defended with one or more of the "broken" preprocessing-based defenses. Using the strong AutoAttack attack rather than the attacks these defenses' authors had used for evaluation, we can break them on all inputs (0% adversarial accuracy) for both L_2 and L_∞ attacks. This weakness to strong attackers was, for all defenses, first pointed out in the articles that successfully broke them (Athalye et al., 2018; He et al., 2017).

The fact that some defenses are effective against weak attacks like FGSM but not against stronger attacks has two possible explanations. One is that such defenses protect against a subset of all existing perturbations, but not all of them: a stronger attacker can find more subtle perturbations that evade the defense. Another is that defenses don't actually protect against perturbations at all, but merely make them perturbations harder to find for attackers by *obfuscating gradients* (Athalye et al., 2018).

Adversarial sparsity offers a simple way to discriminate between these two situations. We expect a partially effective defense to increase sparsity, but not accuracy. An obfuscation-based defense on the other hand would not improve either of these metrics.

On L_2 attacks, in Table 2 we observe that none of these defenses lead to a significant improvement in sparsity. Any L_2-robustness claim regarding these methods is therefore likely to be relying on spurious obfuscation effects. Results are however different on L_∞ attack. While Feature Squeezing does not increase sparsity, JPEG compression and spatial smoothing do ($+20.9$ and $+18.7$ respectively). These defenses are therefore effective against some perturbations. Moreover, feature smoothing does boost robustness when combined with Spatial Smoothing, which was one of the claims of Xu et al. (2018). We argue that the recent claims that most proposed defenses rely solely on subtle obfuscation effects should be revised. In a few examples, we have shown that these defenses offer non-negligible protection against some perturbations. Even if past works have identified obfuscation effects in these defenses using different methods, such as adaptive evaluation (Athalye et al., 2018), they do not explain all of their effect on robustness. Sparsity is complementary to adaptive attacks when evaluating defenses.

6.4. Testing data augmentation on standard models

In Table 3 we report the adversarial sparsity of various models trained with data augmentation. We observe that each of them increases sparsity on both L_2 and L_∞ perturbations. The only exception is CutMix (Yun et al., 2019).

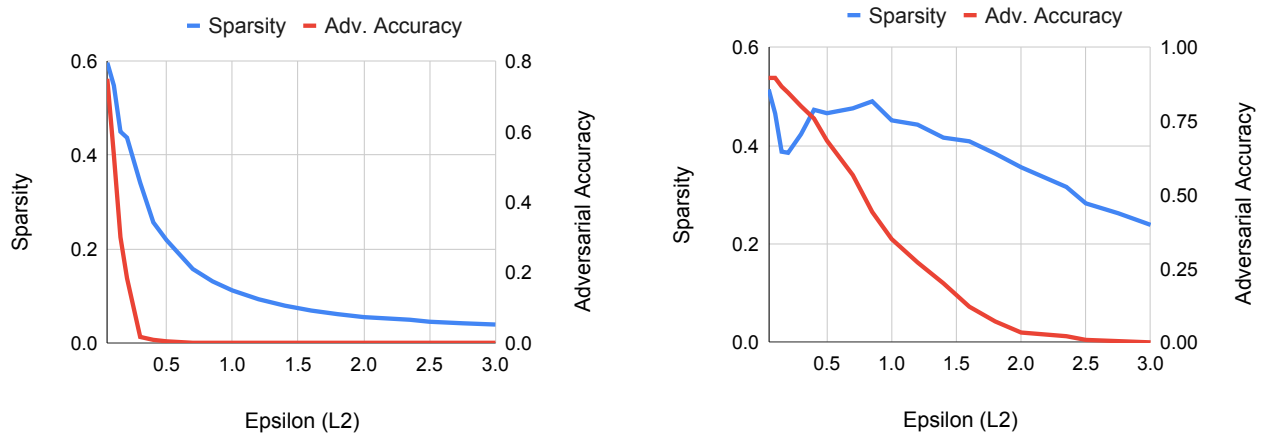
Interestingly CutMix is also the best augmentation for robustness when combined with adversarial training, according to Rebuffi et al. (2021). Moreover, RICAP, which largely outperforms all other augmentations for robustness with standard training in our experiments, is the worst-performing one for adversarial training! A possible explanation is that despite improving robustness, data augmentation interferes with adversarial training: the best augmentations for robustness won't necessarily combine harmoniously with the best training schemes. This explanation is consistent with past works which had shown that augmentation leads to no robustness improvement (Rice et al., 2020). One of the contributions of Rebuffi et al. (2021) was to overcome this issue with methods like weight averaging.

Our results suggest that the potential of RICAP augmentation for robustness is still underexplored. Given its considerable effect on the distribution of adversarial examples, it is likely that a version of RICAP+Adversarial training could outperform CutMix+Adversarial training if the aforementioned interference effects are overcome.

6.5. Influence of the attack radius

Another interesting application of sparsity is to provide a smoother robustness metric than accuracy. We illustrate this in Figure 3a, where we plot both sparsity and accuracy for L_2 attacks for increasing values of the attack radius ϵ, and a standard ResNet18. We observe that while accuracy drops to zero after a small value of ϵ, sparsity decreases much more slowly, only converging to 0 asymptotically.

This property lets us compare models on a much larger range of perturbations. In Figure 3b we plot the equivalent plot for an adversarially trained ResNet18. When comparing Figures 2a and 2b, on large ϵ values accuracy provides little information, being null or close to null for both models. On the other hand, we easily observe that sparsity is over five



(a) Standard ResNet18 model

(b) Adversarially trained model (training radius 0.5)

Figure 3. Evolution of L_2 sparsity and adversarial accuracy of ResNet18 models as a function of the attack radius ϵ

times higher for the adversarially trained model. This shows that training models adversarially with a given attack radius is beneficial even for much larger radii.

In Figure 3b we also observe an interesting fluctuation in the sparsity curve. Although sparsity overall decreases when the attack radius increases overall, the training radius $\epsilon = 0.5$ appears to be a local sparsity maximum. One interpretation is that for some points, the effect of adversarial training is to remove adversarial perturbations from the unit hypersphere, but not inside it. This leads to a strange effect where the model gets a bit more vulnerable when decreasing the attack radius. Because adversarial perturbations in sparsity are restricted to points on the hypersphere, it is particularly affected.

In Appendix D we provide equivalent plots for L_∞ attacks and L_∞ adversarial training, with very similar observations.

7. Conclusion

We have proposed a novel robustness metric named adversarial sparsity. We have shown that it is complementary to adversarial accuracy, offering additional insight into both weak and strong defenses. By applying it to data-augmented models we have found evidence suggesting that some augmentation methods still retain an untapped potential to increase robustness.

While accuracy (certified or not) should likely remain the primary metric for benchmarking strong defenses, we believe that using finer metrics in the research process can benefit the research field. An important direction for future work could be to extend sparsity to threat models beyond norm-bounded perturbations, such as human-perception-based attacks or common corruptions.

References

Athalye, A., Carlini, N., and Wagner, D. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, 2018. URL https://arxiv.org/abs/1802.00420.

Augustin, M., Meinke, A., and Hein, M. Adversarial robustness on in- and out-distribution improves explainability. In *ECCV*, 2020.

Carlini, N. and Wagner, D. Defensive distillation is not robust to adversarial examples, 2016a. URL https://arxiv.org/abs/1607.04311.

Carlini, N. and Wagner, D. A. Towards evaluating the robustness of neural networks. *CoRR*, 2016b.

Choquette-Choo, C. A., Tramer, F., Carlini, N., and Papernot, N. Label-only membership inference attacks. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 1964–1974. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/choquette-choo21a.html.

Cohen, J. M., Rosenfeld, E., and Kolter, J. Z. Certified adversarial robustness via randomized smoothing. *CoRR*, 2019.

Croce, F. and Hein, M. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, 2020.

Croce, F., Andriushchenko, M., Sehwag, V., Debenedetti, E., Flammarion, N., Chiang, M., Mittal, P., and Hein, M. Robustbench: a standardized adversarial robustness

benchmark. In *NeurIPS Datasets and Benchmarks Track*, 2021. URL https://openreview.net/forum?id=SSKZPJCt7B.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*, abs/2010.11929, 2021.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples, 2014.

Göpfert, J. P., Artelt, A., Wersing, H., and Hammer, B. Adversarial attacks hidden in plain sight. In Berthold, M. R., Feelders, A., and Krempl, G. (eds.), *Advances in Intelligent Data Analysis XVIII*, pp. 235–247, Cham, 2020. Springer International Publishing. ISBN 978-3-030-44584-3.

Gowal, S., Qin, C., Uesato, J., Mann, T., and Kohli, P. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *arXiv preprint arXiv:2010.03593*, 2020. URL https://arxiv.org/pdf/2010.03593.

Guo, C., Rana, M., Cissé, M., and van der Maaten, L. Countering adversarial images using input transformations. *CoRR*, abs/1711.00117, 2017. URL http://arxiv.org/abs/1711.00117.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90.

He, W., Wei, J., Chen, X., Carlini, N., and Song, D. Adversarial example defenses: Ensembles of weak defenses are not strong. In *Proceedings of the 11th USENIX Conference on Offensive Technologies*, WOOT'17, pp. 15, USA, 2017. USENIX Association.

Huang, H., Wang, Y., Erfani, S. M., Gu, Q., Bailey, J., and Ma, X. Exploring architectural ingredients of adversarially robust deep neural networks. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=OdklztJBBYH.

Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., and Madry, A. Adversarial examples are not bugs, they are features. 2019.

Khoury, M. and Hadfield-Menell, D. On the geometry of adversarial examples. *ArXiv*, abs/1811.00525, 2018.

Kolter, J. Z. and Wong, E. Provable defenses against adversarial examples via the convex outer adversarial polytope. *CoRR*, 2017.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

Li, S. Concise formulas for the area and volume of a hyperspherical cap. *Asian Journal of Mathematics & Statistics*, 4:66–70, 2011.

Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., and Song, L. Sphereface: Deep hypersphere embedding for face recognition. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6738–6746, 2017.

Ma, X., Li, B., Wang, Y., Erfani, S. M., Wijewickrema, S., Schoenebeck, G., Houle, M. E., Song, D., and Bailey, J. Characterizing adversarial subspaces using local intrinsic dimensionality. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=B1gJ1L2aW.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In *ICLR 2018, Conference Track Proceedings*, 2018.

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

Pang, T., Yang, X., Dong, Y., Xu, K., Zhu, J., and Su, H. Boosting adversarial training with hypersphere embedding. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 7779–7792. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/5898d8095428ee310bf7fa3da1864ff7-Paper.pdf.

Papernot, N., McDaniel, P. D., Wu, X., Jha, S., and Swami, A. Distillation as a defense to adversarial perturbations against deep neural networks. *CoRR*, 2015.

Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., and Swami, A. The Limitations of Deep Learning in Adversarial Settings. In *IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 372–387, 2016. URL https://beerkay.github.io/papers/Berkay2016DLLimitationEuroSP.pdf.

Rade, R. and Moosavi, S.-M. Helper-based adversarial training: Reducing excessive margin to achieve a better accuracy vs. robustness trade-off. In *ICML 2021 Workshop*

on Adversarial Machine Learning, 2021. URL https://openreview.net/forum?id=BuD2LmNaU3a.

Raghu, M., Unterthiner, T., Kornblith, S., Zhang, C., and Dosovitskiy, A. Do vision transformers see like convolutional neural networks? *CoRR*, abs/2108.08810, 2021. URL https://arxiv.org/abs/2108.08810.

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

Rice, L., Wong, E., and Kolter, J. Z. Overfitting in adversarially robust deep learning. In *ICML*, 2020.

Robey, A., Chamon, L., Pappas, G. J., and Hassani, H. Probabilistically robust learning: Balancing average and worst-case performance. In *International Conference on Machine Learning*, pp. 18667–18686. PMLR, 2022.

Salman, H., Ilyas, A., Engstrom, L., Kapoor, A., and Madry, A. Do adversarially robust imagenet models transfer better? In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 3533–3545. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/24357dd085d2c4b1a88a7e0692e60294-Paper.pdf.

Samangouei, P., Kabkab, M., and Chellappa, R. Defense-gan: Protecting classifiers against adversarial attacks using generative models. In *ICLR*, 2018.

Sehwag, V., Mahloujifar, S., Handina, T., Dai, S., Xiang, C., Chiang, M., and Mittal, P. Robust learning meets generative models: Can proxy distributions improve adversarial robustness? *ICLR Workshop on Security and Safety in Machine Learning Systems*, 2021. URL https://arxiv.org/abs/2104.09425.

Shafahi, A., Huang, W. R., Najibi, M., Suciu, O., Studer, C., Dumitras, T., and Goldstein, T. Poison frogs! targeted clean-label poisoning attacks on neural networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pp. 6106–6116, Red Hook, NY, USA, 2018. Curran Associates Inc.

Shao, R., Shi, Z., Yi, J., Chen, P., and Hsieh, C. On the adversarial robustness of visual transformers. *CoRR*, abs/2103.15670, 2021. URL https://arxiv.org/abs/2103.15670.

Shin, R. and Song, D. Jpeg-resistant adversarial images. In *NIPS 2017 Workshop on Machine Learning and Computer Security*, 2017.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. In *ICLR*, 2014. URL http://arxiv.org/abs/1312.6199.

Tramèr, F., Papernot, N., Goodfellow, I., Boneh, D., and Mcdaniel, P. The space of transferable adversarial examples. *ArXiv*, abs/1704.03453, 2017.

Wong, E., Rice, L., and Kolter, J. Z. Fast is better than free: Revisiting adversarial training. *CoRR*, abs/2001.03994, 2020. URL https://arxiv.org/abs/2001.03994.

Xie, C., Wang, J., Zhang, Z., Ren, Z., and Yuille, A. Mitigating adversarial effects through randomization. *arXiv:1711.01991*, 2017.

Xu, W., Evans, D., and Qi, Y. Feature squeezing: Detecting adversarial examples in deep neural networks. In *25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, February 18-21, 2018*. The Internet Society, 2018. URL http://wp.internetsociety.org/ndss/wp-content/uploads/sites/25/2018/02/ndss2018_03A-4_Xu_paper.pdf.

Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., and Yoo, Y. J. Cutmix: Regularization strategy to train strong classifiers with localizable features. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6022–6031, 2019.

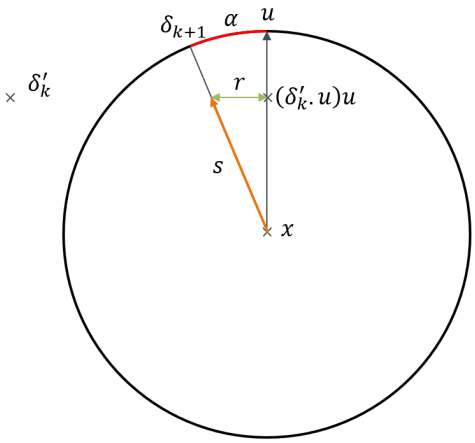


Figure 4. Illustration of the angular projection steps on a 2-dimensional sphere, with $\epsilon = 1$. δ'_k is the perturbation before projection, u is the direction of the cap, and δ_{k+1} is the final perturbation, projected both in norm and angle.

A. Projection on a spherical cap

The computation of L_2 sparsity requires us to replace the projection step in PGD with a projection on a spherical cap (section 4.2). This projection consists of the following steps:

$$p \leftarrow \delta'_k - \delta'_k \cdot u.u \tag{17}$$

$$r \leftarrow min(\frac{\|\delta'_k\|_2}{\|p\|_2} \sin\alpha, 1) \tag{18}$$

$$s \leftarrow \delta'_k \cdot u.u + r.p \tag{19}$$

$$\delta_{k+1} \leftarrow \frac{\min(\epsilon, \|\delta'_k\|_2)}{\|s\|_2} s \tag{20}$$

The output of Eq 19 is a linear combination of δ'_k and u (with positive coefficients) whose angle with u is $\min(\widehat{\delta'_k, u}, \alpha)$. We then rescale this vector as in standard PGD to obtain δ_{k+1}. In Figure 4 we visually illustrate how this procedure returns the closest perturbation to δ'_k of angle at most α with u.

B. Time complexity of sparsity computation

On CIFAR10, computing adversarial sparsity around an input point with 100 directions, 10 search steps, and 20 PGD iterations takes a few seconds for a ResNet-18 model on an Nvidia RTX 2080 Ti. For reference, we find it shorter than running the AutoPGD attack (Croce & Hein, 2020) with default hyperparameters in its worst-case scenario (i.e. when there is no adversarial example).

B.1. Dependence on input dimension

Everything else equal, the duration of sparsity computation in a given direction is proportionate to the duration of one backward pass on the model. It is therefore not significantly longer on ImageNet than on CIFAR10 when the same model architecture is used, the only difference being the input size. Of course, reaching good performance on ImageNet requires (to this day) larger models than on CIFAR10: the current state-of-the-art models have about 2B parameters (Dosovitskiy et al., 2021), while our largest models in this work have less than 100M. This would impact the practical cost of sparsity computation.

One could also think that the number of samples required to confidently evaluate sparsity is much larger in higher dimensions. However, our algorithm does not require to sample in "every" direction, or enough to break the "curse of dimensionality". Sparsity in practice has low variance with respect to the direction (see Section 6.1). Therefore only a few samples are required to estimate with a reasonable margin of error.

B.2. Improving the complexity of sparsity

While attacks over multiple directions can be computed in batches, binary search constitutes the major bottleneck of this algorithm. Some heuristics could help speed up sparsity computation. For example, we could use batched n-Ary search with fewer directions to find a first approximation of sparsity, then confirm/refine it with more directions. As we have mostly worked with reasonably sized models and inputs we have not experimented with such heuristics and leave them as future work.

C. Additional experiments

C.1. ImageNet

L_2 attacks on ImageNet are not as popular as on CIFAR10. We however apply them, both for reference and to verify that angular sparsity computation scales reasonably to a larger input size. We evaluate pretrained ResNet models provided in the Robustbench framework. One is trained in a standard fashion and the other adversarially against L_∞ perturbations, following (Salman et al., 2020). All take inputs of size 224x224. We apply L_2 perturbations of radius $\epsilon = 1.0$. Everything else is similar to the CIFAR10 experiments.

We report the results in Figure 4. We report higher robustness (under all metrics) than we did for CIFAR10; this is consistent with the fact that $\epsilon = 1.0$ in a 224x224 vector space allows a smaller perturbation per pixel budget than $\epsilon = 0.5$ in a 32x32 space. In the absence of L_2-robust, easily available ImageNet models we chose a small value of

1	Architecture	Defenses	Natural accuracy	Adv. accuracy	Sparsity
2	ResNet-50	None	87.8%	0%	0.194
3	ResNet-18	(Salman et al., 2020)	52.9%	39.5%	0.426
4	ResNet-50	(Salman et al., 2020)	64.0%	53.4%	0.415

Table 4. Evaluation of defended and undefended ImageNet models under L_2 attack with 20 iterations and $\epsilon = 1.0$. We report Natural accuracy (without attack), adversarial accuracy (AutoPGD), and adversarial angular (residual) sparsity.

ϵ.

C.2. Visual Transformers

We evaluate a B-16 Visual Transformer Architecture (ViT), pretrained on ImageNet and fine-tuned on CIFAR10 (Dosovitskiy et al., 2021). Recent works have shown that these models learn different features than Convolutional Neural Networks (Raghu et al., 2021) which raises the question of their behavior against adversarial examples. In fact, some works have already claimed that ViTs are more robust to attacks than CNNs (Shao et al., 2021). Investigating these claims is an interesting use case of adversarial sparsity.

We evaluate this ViT model on L_2 perturbations with the same parameters as in Section 6. It reaches a natural accuracy of 97%, greater than any of our ResNet-18 models, and an adversarial accuracy of 0%. Its L_2 sparsity is 0.183, almost equal to that of the ResNet-18 model trained with data augmentation (line 3). The training recipe we use to fine-tune the model employs similar augmentation methods. This suggests that ViTs and ResNets behave similarly against the same L_2 PGD threat model. This challenges the conclusions of (Shao et al., 2021) on the adversarial robustness of ViTs.

C.3. Point-wise variance

When varying the input point there are significant variations in the value of the metrics. Taking the vanilla model evaluated over 100 inputs, for L_2 (resp. L_∞) sparsity we observe a standard deviation of 0.144 (resp. 52.8) with respect to data points. Keeping in mind the considerations in Section 3.3, this hints that the adversarial set is considerably larger for some points than others. The deviation is even larger for adversarially trained models. This is clearly visible in Figure 5, where we plot the histogram of sparsity values of standard and adversarially trained ResNet18 models.

Interestingly, there seems to be a correlation between sparsity on the vanilla model, and accurate prediction on a robust (adversarially trained) model. Using a threshold-based classifier on the vanilla model sparsity to predict whether the robust classifier predicts them correctly, we can reach a precision of 67% at Equal Error Rate. This demonstrates an additional property of adversarial sparsity: to discriminate points that are "easy" to learn with a robust decision boundary (the points with a sparse adversarial set) from harder ones.

D. Influence of attack radius on L_∞ sparsity

Figures 6a and 6b illustrate the evolution of L_∞ sparsity as a function of the attack radius, for a standard and adversarially trained model respectively.

E. A Theoretical Setting for sparsity

We now formalize and prove the results mentioned in section 3.3 linking adversarial sparsity to the number of perturbations. f,x, ϵ and $n \geq 3$ are fixed.

Proposition E.1. *Let μ be the probability measure associated with the uniform distribution over admissible set Δ. Let $\mathcal{D}$ a distribution of sequences of subsets $\Delta^m \subset \Delta$, indexed on M. Assume $\mathcal{D}$ is such that the volume of Δ^m only depends on m:*

$$\frac{\mu(\Delta^m)}{\Delta} = g(m)$$

Assume $Adv(f, x, \epsilon) = \{\delta_j, 1 \leq j \leq k\}$ with (δ_j) uniformly sampled iid. over Δ Then we have:

$$\mathbb{E}_{(\delta_j)\sim\mathcal{U}(\Delta)}[AS(f, \epsilon, x)] = \int_M (1 - g(m))^k dm$$

Proof. Let us note $X_j = \inf\{m, \delta_j \in \Delta^m\}$, such that

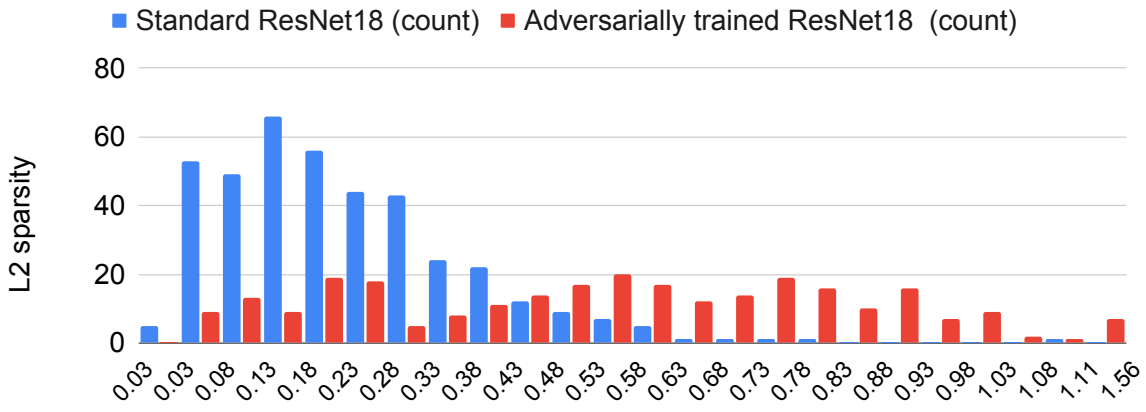
$$\text{AS}(f, x, \epsilon, (\Delta^m)) = \inf_{1\leq j\leq k} X_j$$

Recall that:

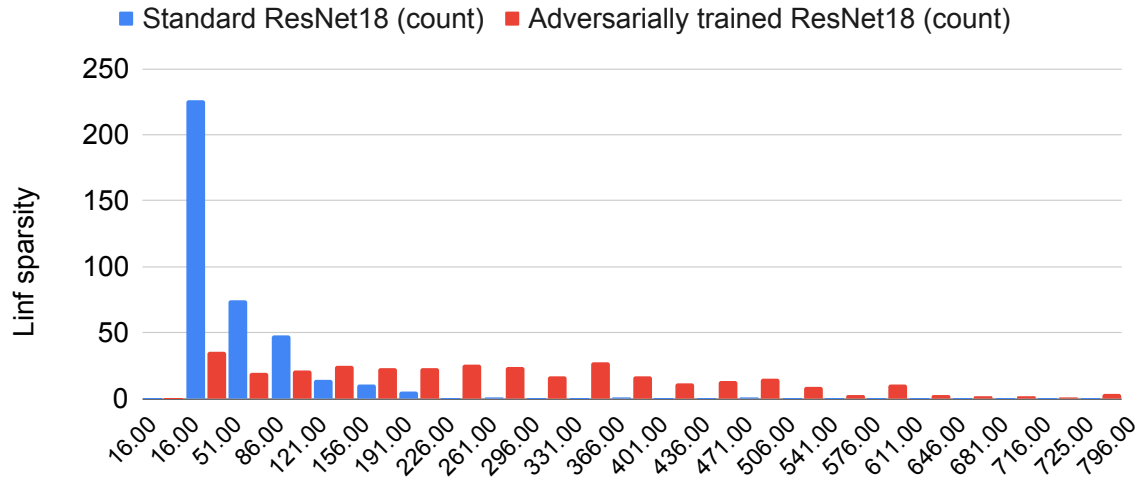
$$\begin{aligned}
&\mathbb{E}_{(\delta_j)\sim\mathcal{U}(\Delta)}[\text{AS}(f, \epsilon, x)] \\
&= \mathbb{E}_{(\delta_j)\sim\mathcal{U}(\Delta)}[\mathbb{E}_{(\Delta^m)\sim\mathcal{D})}[\text{AS}(f, \epsilon, x, (\Delta^m))]] \\
&= \mathbb{E}_{(\Delta^m)\sim\mathcal{D})}[\mathbb{E}_{(\delta_j)\sim\mathcal{U}(\Delta)}[\text{AS}(f, \epsilon, x, (\Delta^m))]] \\
&= \mathbb{E}_{(\Delta^m)\sim\mathcal{D})}[\int_M \mathbb{P}_{(\delta_j)\sim\mathcal{U}(\Delta)}[\text{AS}(f, \epsilon, x, (\Delta^m)) > m]dm]
\end{aligned} \tag{21}$$

Note that

$$\begin{aligned}
\mathbb{P}_{(\delta_j)\sim\mathcal{U}(\Delta)}[X_j > m] &= \mathbb{P}_{(\delta_j)\sim\mathcal{U}(\Delta)}[\delta_j \notin \Delta^m] \\
&= \frac{\mu(\bar{\Delta^m})}{\mu(\Delta)} = 1 - g(m)
\end{aligned} \tag{22}$$

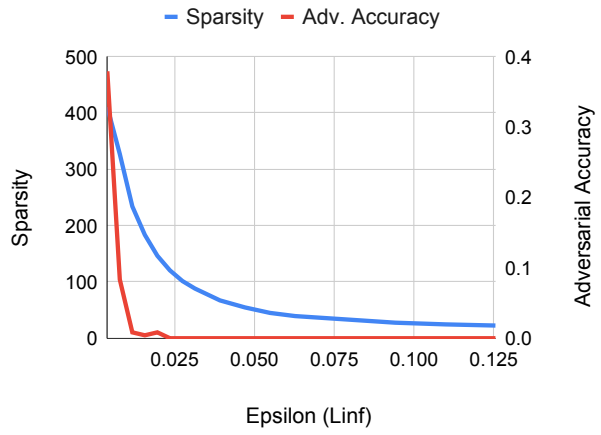


(a) L_2 sparsity

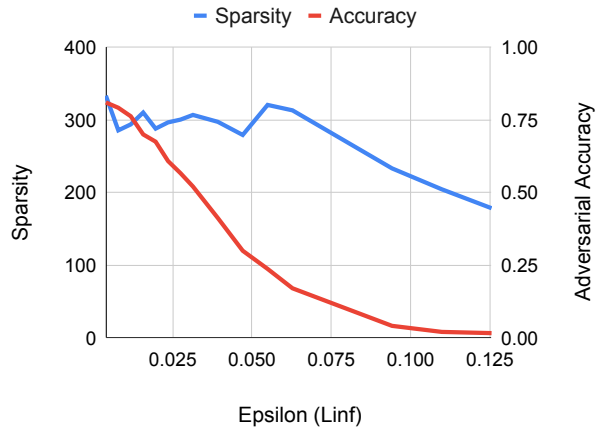


(b) L_∞ sparsity

Figure 5. Histogram of sparsity values for standard and adversarially trained ResNet18 models. It illustrates that sparsity variance with respect to input points is very large, especially for adversarially trained models.



(a) Standard ResNet18 model



(b) Adversarially trained model (training radius .0314)

Figure 6. Evolution of L_∞ sparsity and adversarial accuracy of ResNet18 models as a function of the attack radius ϵ

Moreover, since $X_1, ..., X_k$ are independent,

$$\begin{aligned}&\mathbb{P}_{(\delta_j)\sim\mathcal{U}(\Delta)}[(\inf_{1\leq j\leq k} X_j) > m] \\ &= \mathbb{P}_{(\delta_j)\sim\mathcal{U}(\Delta)}[X_1 > m \wedge ... \wedge X_k > m] \\ &= \Pi_{1\leq j\leq k}\mathbb{P}_{(\delta_j)\sim\mathcal{U}(\Delta)}[\delta_j \notin \Delta^m] \\ &= (1-g(m))^k\end{aligned} \tag{23}$$

It follows

$$\begin{aligned}&\mathbb{E}_{(\delta_j)\sim\mathcal{U}(\Delta)}[\text{AS}(f,\epsilon,x)] \\ &= \mathbb{E}_{(\Delta^m)\sim\mathcal{D})}[\int_M (1-g(m))^k dm] \\ &= \int_M (1-g(m))^k dm\end{aligned} \tag{24}$$

□

In the L_2 case $M = [0, \pi]$, $\Delta = S_2^n$ and Δ^α is a spherical cap of angle α. When $\alpha \leq \frac{\pi}{2}$ formula expressing the area of a spherical cap is derived in (Li, 2011) and equal to:

$$A(\alpha) = \frac{\pi^{\frac{n}{2}}}{\Gamma(\frac{n}{2})} I_{\sin\alpha^2}(\frac{n-1}{2}, \frac{1}{2}) \tag{25}$$

where Γ is the Gamma function and I is the regularized incomplete beta function. Given that $I_1(a, b) = 1$ it follows that:

$$g(m) = g(\alpha) = t_\alpha := I_{\sin\alpha^2}(\frac{n-1}{2}, \frac{1}{2}) \tag{26}$$

.

When $\frac{\pi}{2} < \alpha < \pi$ the cap of angle α is merely the complementary set on the sphere of the cap of angle $\pi - \alpha$. Therefore $g(\alpha) = 1 - g(\pi - \alpha)$ and:

$$\begin{aligned}\mathbb{E}[\text{AS}(f,x,\epsilon)] &= \int_0^\pi ((1-g(\alpha))^k d\alpha \\ &= \int_0^{\frac{\pi}{2}} ((1-t_\alpha)^k + t_\alpha^k) d\alpha\end{aligned} \tag{27}$$

In the L_∞ case, $M = \{0, ..., n\}$, $\Delta = \{\pm 1\}^n$, $\Delta^m = \{\delta \in \Delta \mid \forall q > m \quad \delta_{\sigma(q)} = u_{\sigma(q)}\}$. Therefore $g(m) = 2^{m-n}$ and

$$\mathbb{E}[\text{AS}(f,\epsilon,x)] = \sum_{m=0}^{n} (1-2^{m-n})^k = \sum_{m=0}^{n} (1-2^{-m})^k$$

We now show that $\mathbb{E}[\text{AS}(f,\epsilon,x)] \in \Theta(n - \log_2 k)$. On the one hand,

$$ln((1-2^{-m})^k = k.ln(1-2^{-m}) \leq -k.2^{-m}$$

from which it follows:

$$\begin{aligned}\mathbb{E}[\text{AS}(f,\epsilon,x)] &\leq \sum_{m=0}^{\lfloor \log_2 k \rfloor} e^{-2^{\log_2 k - m}} + \sum_{m=\lfloor \log_2 k \rfloor + 1}^{n} (1-2^{-m})^k \\ &\leq e^{-1} + ... + e^{-m} + (n - \lfloor \log_2 k \rfloor).1 \\ &\leq n - \log_2 k + 1 + \frac{1}{e-1} \\ &= n - \log_2 k + \frac{e}{e-1}\end{aligned} \tag{28}$$

On the other hand,

$$k.ln(1-\frac{1}{k}) \geq k.(1-\frac{1}{1-\frac{1}{k}}) = \frac{-k}{k-1}$$

Thus for $k \geq 2$

$$\begin{aligned}\mathbb{E}[\text{AS}(f,\epsilon,x)] &\geq \sum_{m=\lfloor \log_2 k \rfloor + 1}^{n} (1-2^{-m})^k \\ &\geq (n - \lfloor \log_2 k \rfloor).(1-2^{-\lfloor \log_2 k \rfloor - 1})^k \\ &\geq (n - \log_2 k).(1-\frac{1}{k})^k \\ &\geq (n - \log_2 k).e^{-\frac{k}{k-1}} \\ &\geq (n - \log_2 k).e^{-2}\end{aligned} \tag{29}$$

Since $(1-\frac{1}{2})^2 = \frac{1}{4}$,$(1-\frac{1}{3})^3 = \frac{8}{27}$ and for $k \geq 4$ $e^{=\frac{k}{k-1}} \geq e^{\frac{4}{3}} > \frac{1}{4}$ we can conclude:

$$\mathbb{E}[\text{AS}(f,\epsilon,x)] \geq \frac{(n - \log_2 k)}{4}$$

This also applies when $k = 1$, since $\mathbb{E}[\text{AS}(f,\epsilon,x)] = n - \sum_{m \leq n} 2^{-m} \geq n - 2 \geq \frac{n}{4}$