Implications of Solution Patterns on Adversarial Robustness

Hengyue Liang
Department of Electrical & Computer Engineering
University of Minnesota
liang656@umn.edu

Buyun Liang, Ju Sun
Department of Computer Science & Engineering
University of Minnesota
{liang664, jusun}@umn.edu

Ying Cui
Department of Industrial & Systems Engineering
University of Minnesota
yingcui@umn.edu

Tim Mitchell
Department of Computer Science
Queens College, City University of New York
tim.mitchell@qc.cuny.edu

Abstract

Empirical robustness evaluation (RE) of deep learning models against adversarial perturbations involves solving non-trivial constrained optimization problems. Recent work has shown that these RE problems can be reliably solved by a general-purpose constrained-optimization solver, `PyGRANSO` *with* ***C****onstraint-****F****olding (PWCF). In this paper, we take advantage of PWCF and other existing numerical RE algorithms to explore distinct solution patterns in solving RE problems with various combinations of losses, perturbation models, and optimization algorithms. We then provide extensive discussions on the implications of these patterns on current robustness evaluation and adversarial training. A comprehensive version of this work can be found in [19].*

1. Introduction

Formulations of adversarial robustness in constraint optimization problems In visual recognition, deep neural networks (DNNs) are not robust against perturbations that are easily discounted by human perception—either adversarially constructed or naturally occurring [2, 9, 11–13, 15, 28–30]. A popular way to formalize robustness is by finding the worst-case perturbations within a prescribed radius via maximizing the *adversarial loss* [14, 20]:

$$\begin{aligned} &\max_{\boldsymbol{x}'} \ \ell\left(\boldsymbol{y}, f_{\boldsymbol{\theta}}(\boldsymbol{x}')\right) \\ &\text{s.t. } d\left(\boldsymbol{x}, \boldsymbol{x}'\right) \leq \varepsilon \ , \quad \boldsymbol{x}' \in [0,1]^n \ . \end{aligned} \tag{1}$$

Here, $f_{\boldsymbol{\theta}}$ is the DNN under consideration parameterized by $\boldsymbol{\theta}$, $\boldsymbol{x}$ is an image input, $\boldsymbol{x}'$ is a perturbed version of $\boldsymbol{x}$ with an allowable perturbation radius ε measured by the distance metric d, and $\boldsymbol{x}' \in [0,1]^n$ ensures $\boldsymbol{x}'$ to be a valid image (n: the number of pixels). Another formalism of robustness is by *robustness radius*, defined as the minimal level of perturbation(s) so that $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ and $f_{\boldsymbol{\theta}}(\boldsymbol{x}')$ can lead to different predictions, which was first introduced in [28], even earlier than (1):

$$\begin{aligned} &\min_{\boldsymbol{x}'} \ d\left(\boldsymbol{x}, \boldsymbol{x}'\right) \\ &\text{s.t. } \max_{i \neq y} f^i_{\boldsymbol{\theta}}(\boldsymbol{x}') \geq f^y_{\boldsymbol{\theta}}(\boldsymbol{x}') \ , \ \boldsymbol{x}' \in [0,1]^n \ . \end{aligned} \tag{2}$$

Here, y is the true class of $\boldsymbol{x}$ and $f^i_{\boldsymbol{\theta}}$ represents the i^{th} logit output. Early work assumes that the distance metric d in both (1) and (2) are the ℓ_p distance, where $p = 1, 2, \infty$ are popular choices [11, 20]. Recent work has also modeled non-trivial transformations using metrics other than the ℓ_p distances [2, 9, 12, 13, 15, 16, 29, 30], to capture visually realistic perturbations and generate adversarial examples with more varieties.

As for the applications, (1) is usually associated with an attacker in an adversarial setup, where the attacker tries to find an input that is visually similar to $\boldsymbol{x}$ but can fool $f_{\boldsymbol{\theta}}$

to predict incorrectly, as solutions to (1) lead to worst-case perturbations w.r.t the classification loss. Thus, it is popular to perform robustness evaluation (RE) via (1): generating perturbations over a validation dataset and reporting the classification accuracy (a.k.a. robust accuracy) using the perturbed samples. (1) also motivates adversarial training (AT) to try to achieve adversarial robustness (AR) [11,14,20] (which seems unlikely; see Sec. 4.2):

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}} \max_{\boldsymbol{x}'\in\Delta(\boldsymbol{x})} \ell\left(\boldsymbol{y}, f_{\boldsymbol{\theta}}(\boldsymbol{x}')\right) , \tag{3}$$

where $\mathcal{D}$ is the data distribution, and $\Delta(\boldsymbol{x}) = \{\boldsymbol{x}' \in [0,1]^n : d(\boldsymbol{x},\boldsymbol{x}') \leq \varepsilon\}$, in contrast to the classical supervised learning:

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}} \ell\left(\boldsymbol{y}, f_{\boldsymbol{\theta}}(\boldsymbol{x})\right) . \tag{4}$$

As for (2), it is also a popular choice to calculate robust accuracy in RE[1] [6,7,23] as the solutions will also produce fooling samples $\boldsymbol{x}'$. But more importantly, solving (2) can be used to estimate the sample-wise robustness radius—a quantity that can be used to measure the robustness for every input. However, it is common that existing methods for solving (2) emphasize the role of finding $\boldsymbol{x}'$ but overlook the importance of the robustness radius, e.g., [6,23].

Numerical solvers for solving (1) and (2) (1) is popularly solved by the projected gradient descent (PGD) method. The basic update reads:

$$\boldsymbol{x}'_{new} = \mathcal{P}_{\Delta(\boldsymbol{x})}\left(\boldsymbol{x}'_{old} + t\nabla\ell(\boldsymbol{x}'_{old})\right) \tag{5}$$

where $\mathcal{P}_{\Delta(\boldsymbol{x})}$ is the projection operator to the feasible set $\Delta(\boldsymbol{x})$. When $\Delta(\boldsymbol{x}) = \{\boldsymbol{x}' \in [0,1]^n : \|\boldsymbol{x}' - \boldsymbol{x}\|_p \leq \varepsilon\}$ with $p = 1, 2, \infty$, there are efficient analytical methods to derive (or approximate) $\mathcal{P}_{\Delta(\boldsymbol{x})}$. For these cases, applying PGD is convenient. However, for other general ℓ_p and non-ℓ_p distances, analytical projectors mostly cannot be derived and PGD methods do not apply. Currently, one of the state-of-the-art (SOTA) numerical solvers for (1) is APGD (Auto-PGD) from the `AutoAttack` [7] package. As for solving (2), the difficulty lies in dealing with the highly nonlinear constraint: $\max_{i\neq y} f^i_{\boldsymbol{\theta}}(\boldsymbol{x}') \geq f^y_{\boldsymbol{\theta}}(\boldsymbol{x}')$. One of the SOTA methods, Fast Boundary Attack (FAB) [6], uses iterative linearization to circumvent it. FAB linearizes the constraint at each step, leading to simple solutions to the projection (onto the intersection of the linearized decision boundary and the $[0,1]^n$ box) for particular choices of d (i.e., ℓ_1, ℓ_2 and ℓ_∞ distances). Similar to PGD, FAB does not handle general ℓ_p metrics and non-ℓ_p metrics due to the use of analytical projection. Recent work [18] has proposed to solve (1) using a general-purpose nonlinear optimization solver, `PyGRANSO` **w**ith-**C**onstraint-**F**olding (PWCF). PWCF not only solves (1) effectively with ℓ_1, ℓ_2 and ℓ_∞ distances, but can also handle general ℓ_p ($p \geq 1$) distances and more beyond, such as perceptual distances (PD) [16]. We also find that PWCF can easily adapt to solving (2) based on the techniques introduced in [18].

[1]One can also perform AT using (2) via bi-level optimization; see, e.g., [33].

Our contributions In this paper, we consider the existing numerical methods (APGD, FAB, and PWCF) for (1) and (2). We compare the sparsity patterns of their solutions using different solvers, losses, and distance metrics. And then we discuss the implications on RE and AT:

1. Different combinations of distance metrics d, losses ℓ, and optimization solvers used to solve (1) and (2) can induce different sparsity patterns in the solutions found. In terms of computing robust accuracy, combining solutions with all possible patterns is important to obtain a reliable and accurate result.

2. Robust accuracy at a preset perturbation level ε used in RE is a bad metric to measure robustness: a model achieves better robust accuracy than other models at one level does not imply that the model can obtain a superior one than other models at other levels; see Fig. 1. in [26]. Instead, performing RE via (2) can be more beneficial, as the sample-wise robustness radius found contains richer information.

3. Due to the pattern difference in solving (1) by different combinations of solvers, losses, and distance metrics, the common practice of AT (3) with PGD on a single distance metric may not be able to achieve generalizable adversarial robustness at all—adversarially trained models may only be robust to the patterns they have seen during training.

Although previous works, e.g., [3, 7, 10], have mentioned the necessity of involving diverse solvers to achieve more reliable RE, our paper is the first to quantify the meaning of diversity in terms of the sparsity patterns to our knowledge. We refer the reader to [19] for a more comprehensive version of this paper.

2. A brief overview of PWCF

A powerful general-purpose constrained optimization solver for deep learning problems General **no**n**l**inear **opt**imization (NLOPT) problems take the form [1]:

$$\begin{aligned} &\min_{\boldsymbol{x}} \; g(\boldsymbol{x}) \\ &\text{s.t. } c_i(\boldsymbol{x}) \leq 0 \; \forall \, i \in \mathcal{I}; \; h_j(\boldsymbol{x}) = 0 \; \forall \, j \in \mathcal{E}, \end{aligned} \tag{6}$$

where $g(\cdot)$ is a continuous objective function; c_i's and h_j's are continuous inequality and equality constraints, respectively. `PyGRANSO`[2] [8, 17] is a recent `PyTorch`-port of the

[2]https://ncvx.org

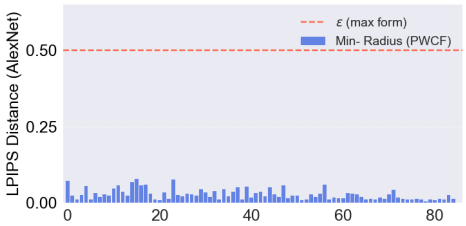


Figure 1. Robustness radius (y-axis) found by solving (2) using PWCF with perceptual distance on 85 ImageNet-100 images. The x-axis are the image indices. The red dashed line is the proposed budget $\varepsilon = 0.5$ used in [16] for PAT (1).

powerful MATLAB package `GRANSO` [8] that can handle general NLOPT problems of form (6) with non-differentiable g, c_i's, and h_j's, equipped with auto-differentiation and GPU acceleration. PWCF blends `PyGRANSO` with constraint-folding and other beneficial techniques, and can handle a wide range of NLOPT problems in deep learning, such as (1) and (2). We refer the readers to [17, 18] for more details.

Solving (1) and (2) with perceptual distance (PD) Here we briefly demonstrate the ability of PWCF in solving (1) and (2), using PD as an example. We consider the following PD [32]:

$$\begin{aligned} & d(\boldsymbol{x}, \boldsymbol{x}') \doteq \|\phi(\boldsymbol{x}) - \phi(\boldsymbol{x}')\|_2 \\ \text{where} \quad & \phi(\boldsymbol{x}) \doteq [\, \widehat{g}_1(\boldsymbol{x}), \ldots, \widehat{g}_L(\boldsymbol{x}) \,] \end{aligned} \tag{7}$$

where $\widehat{g}_1(\boldsymbol{x}), \ldots, \widehat{g}_L(\boldsymbol{x})$ are the vectorized intermediate feature maps from pre-trained DNNs (e.g., AlexNet). [16] first proposed solving (1) with PD at $\varepsilon = 0.5$ as a new form of adversarial attack (termed perceptual attack (PAT)). In [16], three methods are proposed to solve it: 1) Perceptual Projected Gradient Descent (PPGD) which is based on iterative linearization and projection; 2) Lagrangian Perceptual Attack (LPA) which is a penalty method with iterative projections; 3) a variant of LPA without projection (Fast-LPA). For (2), no existing work has considered solving it with PD.

Here, we show in Fig. 1 the robustness radii[3] found by PWCF in solving (2) on 85 ImageNet-100 images. Compared to the preset ε used in [16], much smaller radii for every sample are found by PWCF. We can conclude that: 1) PWCF solves (2) reasonably well; 2) the choice of ε is too large to be a reasonable distance budget in (1). Next, we solve (1) on the ImageNet-100 validation set with $\varepsilon = 0.5$[4], reporting both the attack success rate and the constraint violation rate. According to Fig. 1, the sample-wise robustness radius is much smaller than 0.5, indicating that effective solvers are expected to achieve 100% attack success rate with 0% violations. As shown in Tab. 1, PWCF with margin loss is the only one that meets this standard and is clearly the best.

[3]We use the 'pat_alexnet_0.5.pt' from https://github.com/cassidylaidlaw/perceptual-advex, where the authors tested and showed its ℓ_2- and ℓ_∞- robustness in the original work.

[4]Using the same adversarially pretrained model as in Fig. 1

	cross-entropy loss		margin loss	
Method	**Viol. (%) ↓**	**Att. Succ. (%) ↑**	**Viol. (%) ↓**	**Att. Succ. (%) ↑**
Fast-LPA	73.8	3.54	41.6	56.8
LPA	**0.00**	80.5	**0.00**	97.0
PPGD	5.44	25.5	**0.00**	38.5
PWCF (ours)	0.62	93.6	**0.00**	100

Table 1. Performance comparison of solving (1) with the perceptual distance, using cross-entropy and margin losses, respectively. **Viol.** reports the ratio of final solutions that violate the constraint; **Att. Succ.** is the ratio of all feasible and successful attack samples divided by total number of samples—higher indicates more effective optimization performance.

Figure 2. The 'fish' image example from Imagenet-100 validation set that is used to generate the pattern visualizations in Fig. 3 and Fig. 4.

3. Different combinations of ℓ, d, and the solvers prefer different solution patterns

We now demonstrate that using different combinations of **1)** distance metrics d, **2)** solvers, and **3)** losses ℓ can lead to different sparsity patterns in the following two ways:

1. **Visualization of perturbation images:** we take a 'fish' image (Fig. 2) from the ImageNet-100 validation set, employ various combinations of losses ℓ, distance metrices d and solvers to (1) and (2). Fig. 3 and Fig. 4 visualize the perturbation image $\boldsymbol{x}' - \boldsymbol{x}$, and the histogram of the element-wise error magnitude $|\boldsymbol{x}' - \boldsymbol{x}|$ to display the difference in pattern.

2. **Statistics of sparsity levels:** we use the soft sparsity measure $\|\boldsymbol{x}' - \boldsymbol{x}\|_1 / \|\boldsymbol{x}' - \boldsymbol{x}\|_2$ to quantify the patterns—the higher the value, the denser the pattern. Fig. 5 and Fig. 6 display the histograms of the sparsity levels of the error images derived by solving (1) and (2), respectively. Here, we used a fixed set of 500 ImageNet-100 images from the validation set.

Contrary to the ℓ_1 distance that induces sparsity, ℓ_∞ promotes dense perturbations with comparable entry-wise magnitudes [27] and ℓ_2 promotes dense perturbations whose entries follow power-law distributions. These varying sparsity patterns due to d's are evident when we compare the solutions with the same solver and loss but with different distances, where **1)** the shapes of the histograms in Fig. 4 and the ranges of the values are very different; **2)** the sparsity

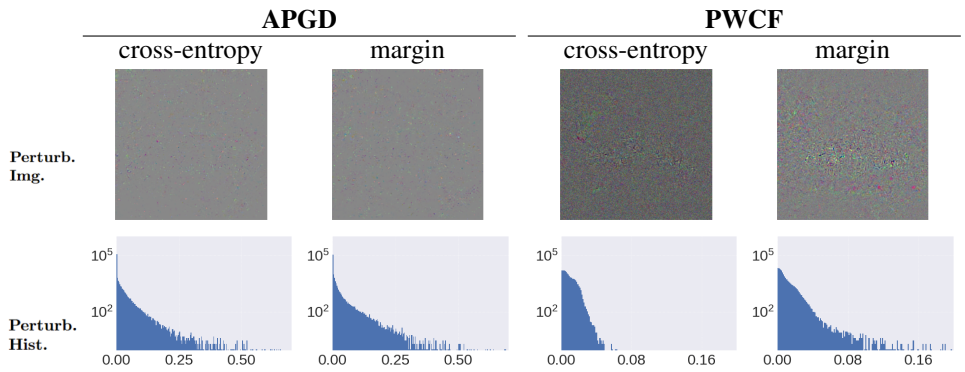


Figure 3. Visualization of perturbation images by solving (1) with ℓ_1 distance, cross-entropy and margin losses, and different solvers (APGD and PWCF). The top are the perturbation images $\boldsymbol{x}' - \boldsymbol{x}$, which have been normalized to the range $[0, 1]$ for better visualization; the bottom are the histograms of the element-wise perturbation magnitude $|\boldsymbol{x}' - \boldsymbol{x}|$, where the x-axes are the absolute pixel values.

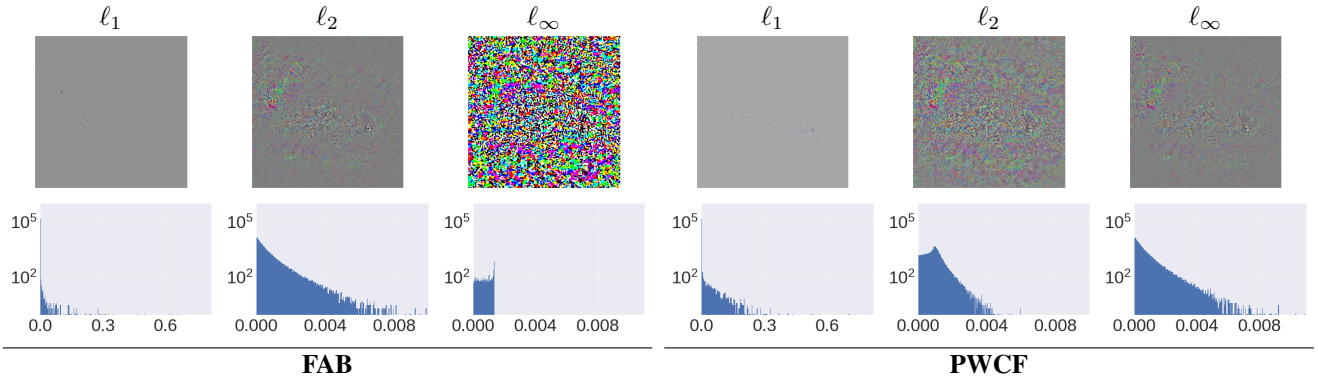


Figure 4. Visualizations of perturbation images ($\boldsymbol{x}' - \boldsymbol{x}$, top row) and the histogram of element-wise perturbation magnitude ($|\boldsymbol{x}' - \boldsymbol{x}|$, bottom row) by solving (2). Note that the comparison between FAB and PWCF may not be as straightforward as Fig. 3 because the radii found by solving (2) are likely different in scale. However, the shape of the histograms can still reveal the pattern differences.

measures show a shift from left to right along the horizontal axis in Fig. 5 and Fig. 6. In addition to d's, we also highlight other patterns induced by the loss ℓ and the solver:

- **Using margin and cross-entropy losses in solving (1) induce different sparsity patterns** Columns 'cross-entropy' and 'margin' of PWCF in Fig. 3 depict the pattern difference with clear divergences in the histograms of error magnitude; for example, the error values of PWCF-cross-entropy are more concentrated towards 0 compared to PWCF-margin. The sparsity measures in Fig. 5 can further confirm the existence of the difference due to the loss used to solve (1), more for PWCF than APGD.
- **PWCF's solutions have more variety in sparsity than APGD and FAB** For the same d and loss used to solve (1), Fig. 5 shows that PWCF's solutions have a wider spread in the sparsity measure than APGD. The same observation can be found in Fig. 6 as well between PWCF and FAB in solving (2).

Here, we provide a conceptual explanation of why different sparsity patterns can occur. We take the ℓ_1 distance (i.e., $\|\boldsymbol{x} - \boldsymbol{x}'\|_1 \leq \varepsilon$) and ignore the box constraint $\boldsymbol{x}' \in [0, 1]$ in (1) as an example. For simplicity, we take $0/1$ classification error $\ell(\boldsymbol{y}, f_{\boldsymbol{\theta}}(\boldsymbol{x}')) = \mathbb{1}\left\{\max_i f^i_{\boldsymbol{\theta}}(\boldsymbol{x}') \neq y\right\}$ as the loss ℓ. Note that ℓ is maximized whenever $f^i_{\boldsymbol{\theta}}(\boldsymbol{x}') > f^y_{\boldsymbol{\theta}}(\boldsymbol{x}')$ for a certain $i \neq y$, so that $\boldsymbol{x}'$ crosses the local decision boundary between the i-th and y-th classes; see Fig. 7 (a). In practice, people set a substantially larger perturbation budget in (1) than the robustness radius, which can be estimated by solving (2)—see Fig. 7 (b). Thus, there can be infinitely many global maximizers (the shaded blue regions in Fig. 7 (a)). As for the patterns, the solutions in the shaded blue region on the left are denser in pattern than the solutions on the top. For other general losses, such as cross-entropy or margin loss, the set of global maximizers might change, but the patterns can possibly be more complicated due to the typically complicated nonlinear decision boundaries associated with DNNs. As for (2), multiple global optimizers and pattern differences can exist as well, but the optimizers share the same radius.

4. Implications from the different patterns

4.1. Current empirical RE may not be sufficient

As introduced in Sec. 1, the most popular empirical RE practice currently is solving (1) with a preset level of ε, using a fixed set of algorithms. Then robust accuracy is reported [4, 22, 25]. Here, we challenge its validity for measuring robustness.

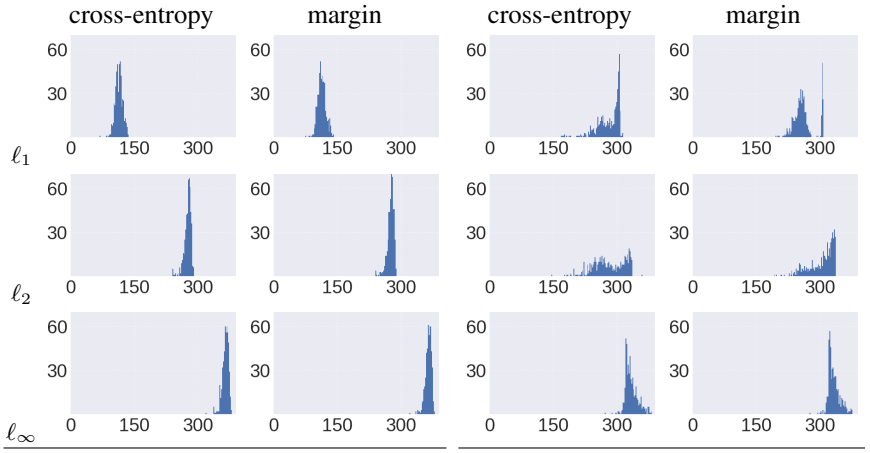


Figure 5. Histograms of the sparsity measure by solving (1) with different ℓ's (cross-entropy and margin losses), different d's (ℓ_1, ℓ_2 and ℓ_∞) and different solvers (APGD and PWCF). For fair comparisons, we use a non-adversarially trained model here so that each x' is a successful adversarial example. With the same d and ℓ, the difference in the distributions of the sparsity measure between APGD and PWCF clearly exists; the sparsity variation is noticeably greater in PWCF than in APGD when the same d and ℓ are used.

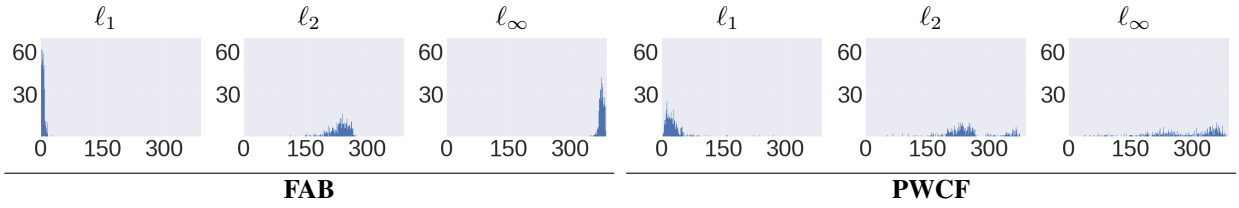
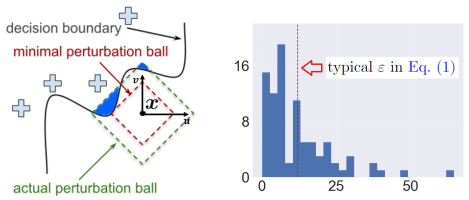


Figure 6. Histograms of the sparsity measure by solving (2) with different d's (ℓ_1, ℓ_2 and ℓ_∞). Under the same solver (FAB or PWCF), the shift of solution patterns from sparse to dense due to d is obvious. Given the same d, the solutions of PWCF have more variety in sparsity than those of FAB, showing the influence of the solver on the solution patterns.



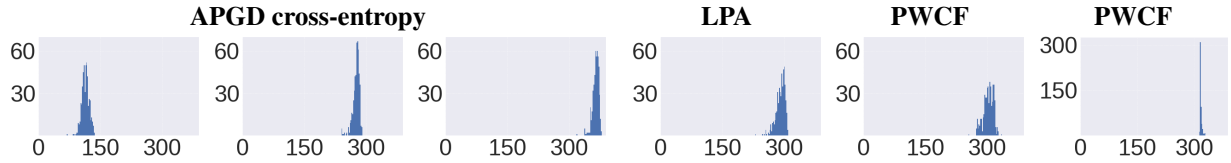
(a) (b)

Figure 7. **(a):** geometry of (1) with multiple global maximizers. $\boldsymbol{u}$ and $\boldsymbol{v}$ are the basis vectors of the 2-dimensional coordinate. Here we consider the ℓ_1 ball around $\boldsymbol{x}$, and ignore the box constraint $\boldsymbol{x}' \in [0,1]^n$. Depending on the loss ℓ used, part or the whole of the blue regions becomes the set of global or near-global maximizers. **(b):** histogram of the ℓ_1 robustness radii estimated by solving (2) for 88 CIFAR-10 images. $\varepsilon = 12$ (red dashed line) is the typical preset perturbation budget used in (1).

Diversity matters for robust accuracy to be trustworthy As shown in Sec. 3, the perturbations found by different numerical methods can have different sparsity patterns. This implies that for robust accuracy to be numerically reliable, including as many solvers to cover as many patterns as possible is necessary. Although works as [3, 7, 10] have mentioned the necessity of diversity in solvers, our paper is the first to quantify such diversity in terms of sparsity patterns from their solutions. However, the existence of infinitely many patterns may be possible. Thus, it is also possible that faithful robust accuracy may not be able to achieve in principle.

Robust accuracy is not a good robustness metric The motivation for using (1) for RE is usually associated with the attacker-defender setting, where solutions ($\boldsymbol{x}'$) are viewed as a test bench for all possible future attacks. Ideally, the DNNs must be robust against all adversarial samples found. However, it is questionable whether robust accuracy faithfully reflects this notion of robustness: **1)** why the commonly used attack budget ε is a reasonable choice needs to be justified. For example, $\varepsilon = 0.03$ is commonly used for the ℓ_∞ distance, e.g., in [4]. We could not find rigorous answers in the previous literature and suspect that the choices are purely empirical. For example, [4] states the motivation as "...the true label should stay the same for each in-distribution input within the perturbation set..." but this claim can also support using other values; **2)** more importantly, Fig. 1. in [26] shows that a model having a higher robust accuracy than other models at one ε level does not imply that such a model is also more robust at other levels. The clean-robust accuracy trade-off [24, 31] may also be interpreted similarly[5]—they are just most robust to different ε levels. Thus, robust accuracy is not a complete and trustworthy measure, and conclusions about robustness drawn from robust accuracy based on a single ε level are misleading. **Robustness ra-**

[5]The *clean-robust accuracy trade-off* refers to the phenomenon where a non-adversarially trained model has the best clean accuracy (at level $\varepsilon = 0$) and the worst robust accuracy (at the commonly used ε), and vise versa for the adversarially trained models.



(a) ℓ_1 **(b)** ℓ_2 **(c)** ℓ_∞ **(d)** PD-ℓ_2 **(e)** PD-ℓ_2 **(f)** PD-ℓ_1

Figure 8. Histograms of the sparsity measure on 500 ImageNet-100 images by solving (1) with different d's (ℓ_p and PD) and different solvers (APGD with cross-entropy loss, LPA and PWCF). LPA-PD-ℓ_2 is the perceptual attack used in [16]. Using the PD in (1) also results in different sparsity patterns due to the solver and the inner ℓ_p distance used (see PD-ℓ_2, PD-ℓ_1 related figures above).

dius is a better robustness measure If our goal is indeed to understand the robustness limit of a given DNN model, solving (2) seems more advantageous, especially that:

1. **Robustness radius is more reliable:** unlike that the pattern differences in solving (1) can lead to unreliable robust accuracy due to possibly multiple solutions, the robustness radius found by solving (2) is not sensitive to the existence of multiple solutions.

2. **Robustness radius is sample adaptive:** in contrast to the rigid perturbation budget ε used in (1), the robustness radius is the sample-wise distances to the closest decision boundaries.

A clear application of the robustness radius is that we can identify hard (less robust) samples for a given model if the corresponding robustness radii are small.

4.2. Adversarial training may not help to achieve generalizable robustness

Solution patterns can explain why ℓ_p robustness does not generalize Despite the effort of finding ways to achieve generalizable AR, it is widely observed that AR achieved by AT does not generalize across simple ℓ_p distances [5,21]. For example, models adversarially trained by ℓ_∞-attacks do not achieve good robust accuracy with ℓ_2-attacks; ℓ_1 seems to be a strong attack for all other ℓ_p distances, and even on itself. Note that [20] has observed that the (approximate) global maximizers are distinct and spatially scattered; the patterns we discussed in Sec. 3 provide a plausible explanation of why AR achieved by AT is expected not to be generalizable—the model just cannot perform well on an unseen distribution (patterns) from what it has seen during training.

Adversarial training with perceptual distances does not solve the generalization issue [16] claims that using PD (Eq. (7)) in (1) can approximate the universal set of adversarial attacks, and models adversarially trained with PAT can generalize to other unseen attacks. However, we challenge the above conclusion: 'unseen attacks' does not necessarily translate to 'novel perturbations', especially if we investigate the patterns:

1. If we test the ℓ_2 and PAT pre-trained models[6] in [16] by APGD-CE-ℓ_1 ($\varepsilon = 1200$) attack (on ImageNet-100 images), both will achieve 0% robust accuracy—PAT pretrained models do not generalize better to ℓ_1 attacks compared with others.

2. By investigating the sparsity patterns similar to Sec. 3, the adversarial perturbations generated by solving (1) with PD are shown to be similar to the APGD-CE-ℓ_2 generated ones, see (a)-(d) in Fig. 8. This may explain why the ℓ_2 and PAT pre-trained models in [16] have comparable robust accuracy against multiple tested attacks.

3. Substituting the ℓ_2 distance by the ℓ_1 distance in Eq. (7) as the new PD:

$$d(\boldsymbol{x}, \boldsymbol{x}') \doteq \|\phi(\boldsymbol{x}) - \phi(\boldsymbol{x}')\|_1 \,, \tag{8}$$

the solution patterns will change; see (e)-(f) in Fig. 8. Furthermore, (d)-(e) in Fig. 8 also shows that different solvers (LPA and PWCF) will also result in different patterns even for PD—PAT will likely suffer from the pattern differences the same way as popular ℓ_p-attacks, thus not being 'universal'.

To conclude, we think that it is unclear so far whether using the perceptual distances in the AT pipeline can be beneficial in addressing the generalization issue in robustness.

5. Summary

In this paper, we briefly introduce PWCF, a general-purpose nonlinear optimization solver that can handle (1) and (2) with general distance metrics. Taking advantage of the existing numerical solvers, we observe that applying different combinations of losses ℓ, distance metrics d and solvers to these two formulations can lead to different sparsity patterns in the solutions. Having provided our explanations on why pattern difference can occur, we then discuss the implications on adversarial robustness: **1)** the current practice of RE based on solving (1) can be insufficient and misleading; **2)** finding the sample-wise robustness radius by solving (2) can be a better robustness metric; **3)** achieving generalizable robustness via the current practice of AT is intrinsically difficult.

[6] Correspond to ℓ_2 and PAT-AlexNet in Table 3 of [16]

References

[1] Dimitri Bertsekas. *Nonlinear Programming 3rd Edition*. Athena Scientific, 2016. 2

[2] Anand Bhattad, Min Jin Chong, Kaizhao Liang, Bo Li, and David A Forsyth. Big but imperceptible adversarial perturbations via semantic manipulation. *arXiv preprint arXiv:1904.06347*, 1(3), 2019. 1

[3] Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. On evaluating adversarial robustness. *arXiv:1902.06705*, Feb. 2019. 2, 5

[4] Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. 4, 5

[5] Francesco Croce and Matthias Hein. Provable robustness against all adversarial ℓ_p-perturbations for $p \geq 1$. *arXiv preprint arXiv:1905.11213*, 2019. 6

[6] Francesco Croce and Matthias Hein. Minimally distorted adversarial examples with a fast adaptive boundary attack. In *International Conference on Machine Learning*, pages 2196–2205. PMLR, 2020. 2

[7] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pages 2206–2216. PMLR, 2020. 2, 5

[8] Frank E Curtis, Tim Mitchell, and Michael L Overton. A bfgs-sqp method for nonsmooth, nonconvex, constrained optimization and its evaluation using relative minimization profiles. *Optimization Methods and Software*, 32(1):148–181, 2017. 2, 3

[9] Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. Exploring the landscape of spatial robustness. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1802–1811. PMLR, 09–15 Jun 2019. 1

[10] Justin Gilmer, Ryan P Adams, Ian Goodfellow, David Andersen, and George E Dahl. Motivating the rules of the game for adversarial example research. *arXiv preprint arXiv:1807.06732*, 2018. 2, 5

[11] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. 1, 2

[12] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2018. 1

[13] Hossein Hosseini and Radha Poovendran. Semantic adversarial examples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1614–1619, 2018. 1

[14] Ruitong Huang, Bing Xu, Dale Schuurmans, and Csaba Szepesvari. Learning with a strong adversary. *arXiv:1511.03034*, Nov. 2015. 1, 2

[15] Cassidy Laidlaw and Soheil Feizi. Functional adversarial attacks. *Advances in neural information processing systems*, 32, 2019. 1

[16] Cassidy Laidlaw, Sahil Singla, and Soheil Feizi. Perceptual adversarial robustness: Defense against unseen threat models. In *ICLR*, 2021. 1, 2, 3, 6

[17] Buyun Liang, Tim Mitchell, and Ju Sun. NCVX: A general-purpose optimization solver for constrained machine and deep learning. 2022. 2, 3

[18] Hengyue Liang, Buyun Liang, Ying Cui, Tim Mitchell, and Ju Sun. Optimization for robustness evaluation beyond ℓ_p metrics. In *OPT 2022: Optimization for Machine Learning (NeurIPS 2022 Workshop)*, 2022. 2, 3

[19] Hengyue Liang, Buyun Liang, Le Peng, Ying Cui, Tim Mitchell, and Ju Sun. Optimization and optimizers for adversarial robustness. *arXiv preprint arXiv:2303.13401*, 2023. 1, 2

[20] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 1, 2, 6

[21] Pratyush Maini, Eric Wong, and Zico Kolter. Adversarial robustness against the union of multiple perturbation models. In *International Conference on Machine Learning*, pages 6640–6650. PMLR, 2020. 6

[22] Nicolas Papernot, Fartash Faghri, Nicholas Carlini, Ian Goodfellow, Reuben Feinman, Alexey Kurakin, Cihang Xie, Yash Sharma, Tom Brown, Aurko Roy, et al. Technical report on the cleverhans v2. 1.0 adversarial examples library. *arXiv preprint arXiv:1610.00768*, 2016. 4

[23] Maura Pintor, Fabio Roli, Wieland Brendel, and Battista Biggio. Fast minimum-norm adversarial attacks through adaptive norm constraints. *Advances in Neural Information Processing Systems*, 34, 2021. 2

[24] Aditi Raghunathan*, Sang Michael Xie*, Fanny Yang, John Duchi, and Percy Liang. Adversarial training can hurt generalization. In *ICML 2019 Workshop on Identifying and Understanding Deep Learning Phenomena*, 2019. 5

[25] Jonas Rauber, Wieland Brendel, and Matthias Bethge. Foolbox: A python toolbox to benchmark the robustness of machine learning models. *arXiv preprint arXiv:1707.04131*, 2017. 4

[26] Kaustubh Sridhar, Souradeep Dutta, Ramneet Kaur, James Weimer, Oleg Sokolsky, and Insup Lee. Towards alternative techniques for improving adversarial robustness: Analysis of adversarial training at a spectrum of perturbations. *arXiv preprint arXiv:2206.06496*, 2022. 2, 5

[27] Christoph Studer, Wotao Yin, and Richard G. Baraniuk. Signal representations with minimum ℓ_∞. In *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, oct 2012. 3

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

[29] Eric Wong, Frank R. Schmidt, and J. Zico Kolter. Wasserstein adversarial examples via projected sinkhorn iterations. *arXiv:1902.07906*, Feb. 2019. 1

[30] Chaowei Xiao, Junyan Zhu, Bo Li, Warren He, Mingyan Liu, and Dawn Song. Spatially transformed adversarial examples. In *International Conference on Learning Representations*, 2018. 1

[31] Yao-Yuan Yang, Cyrus Rashtchian, Hongyang Zhang, Russ R Salakhutdinov, and Kamalika Chaudhuri. A closer look at accuracy vs. robustness. *Advances in Neural Information Processing Systems*, 33:8588–8601, 2020. 5

[32] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 3

[33] Yihua Zhang, Guanhua Zhang, Prashant Khanduri, Mingyi Hong, Shiyu Chang, and Sijia Liu. Revisiting and advancing fast adversarial training through the lens of bi-level optimization. *arXiv:2112.12376*, Dec. 2021. 2