Adversarial robustness without perturbations

by

Adrián Rodríguez Muñoz

B.S., Mathematics, Universitat Politecnica de Catalunya (UPC), 2022
B.S., Data Science and Engineering, Universitat Politecnica de Catalunya (UPC), 2022

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2024

Authored by:	Adrián Rodríguez Muñoz Department of Electrical Engineering and Computer Science May 17, 2024
Certified by:	Antonio Torralba Professor of Electrical Engineering and Computer Science, Thesis Supervisor
Accepted by:	Leslie A. Kolodziejski Professor of Electrical Engineering and Computer Science Chair, Department Committee on Graduate Students

Adversarial robustness without perturbations

by

Adrián Rodríguez Muñoz

Submitted to the Department of Electrical Engineering and Computer Science
on May 17, 2024 in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

ABSTRACT

Models resistant to adversarial perturbations are stable around the neighbourhoods of input images, such that small changes, known as *adversarial attacks*, cannot dramatically change the prediction. Currently, this stability is obtained with *Adversarial Training*, which directly teaches models to be robust by training on the perturbed examples themselves. In this work, we show the surprisingly similar performance of instead regularizing the model input-gradients of un-perturbed examples only.

Regularizing the input-gradient norm is commonly believed to be significantly worse than Adversarial Training. Our experiments determine that the performance of *Gradient Norm* critically depends on the smoothness of the activation functions of the model, and is in fact highly peformant on modern vision transformers that natively use smooth GeLU over piecewise linear ReLUs. On ImageNet-1K, *Gradient Norm* regularization achieves more than 90% of the performance of state-of-the-art Adversarial Training with PGD-3 (52% vs. 56%) with 60% of the training time and without complex inner-maximization.

Further experiments shed light on additional properties relating model robustness and input-gradients of unperturbed images, such as asymmetric color statistics. Suprisingly, we also show significant adversarial robustness may be obtained by simply conditioning gradients to focus on image edges, without explicit regularization of the norm.

Thesis supervisor: Antonio Torralba
Title: Professor of Electrical Engineering and Computer Science

Acknowledgments

I would first like to thank my supervisor Professor Antonio Torralba for his incredible guidance and insights. His amazing creative vision and optimistic problem-solving are a privilege to study under. Next, I would also like to thank all my labmates and collaborators. In particular, I want to especially acknowledge Tongzhou Wang, who is a co-author on the research described on this manuscript, and without whom the success of this project would have been impossible. His approach to research and science has been amazing to learn from. I would like to thank the "la Caixa" Foundation and CSAIL for the support they have provided for me to perform the resarch of this thesis. I also want to thank my amazing partner Mackenzie Sullivan for her constant joy and encouragement. Finally, I would like to thank my parents and brother for their years of upbringing and endless support which make me who I am.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Deep neural networks have become the gold standard for computer vision tasks. Yet they are also extremely brittle. Adversarial examples [1] are small manipulations to input images that can cause highly-performant models to fail catastrophically. For example, the accuracy of an ImageNet deep classifier drops from 84% to 0% under such attacks, even though the perturbed images look identical to humans.

To safely deploy these models in critical tasks such as medicine or autonomous vehicles, extensive research has been devoted to making robust models that are invariant to these small perturbations. The current foremost paradigm for obtaining robust models in practice has been *Adversarial Training* [2], which trains models in a minimax fashion, optimizing classification losses over the attack-perturbed images. This approach is effective (yielding 60% robust accuracy under attack), but is also extremely computationally expensive, taking 3.92× wallclock time per training iteration compared to normal training. Therefore, it is important to seek properties of robust models that can be optimized much more efficiently.

In this work, we analyze the fundamental differences between robust and non-robust models from the perspective of *loss-input gradients*:

$$\nabla_x \mathcal{L} \coloneqq \underbrace{\nabla_x \mathcal{L}_{\text{CE}}(f_\theta(x), y)}_{\text{loss-input gradient of model } f_\theta \text{ on example } x \text{ with groundtruth class } y}, \tag{1.1}$$

where $\mathcal{L}_{\text{CE}}$ is the cross entropy loss. The quantity $\nabla_x \mathcal{L}$ is related to the first-order Taylor expansion of model loss. It is known that a smaller *Gradient Norm* $\nabla_x \mathcal{L}$ is correlated to model smoothness and thus robustness [5], [6]. Figure 1.1 highlights the substantial visual differences between the loss-input gradients of vulnerable and robust models of the same architecture. Such differences reveal the regions of the image that the model uses to make predictions. Notably, the gradients for robust models are much more image-like. In robust models, the gradients are generally focused on the edges of input images, and in fact, have much smaller magnitudes (Table 4.1). In this thesis, we question if these properties contribute to robustness, or are simply irrelevant byproducts.

Previous research showed that simply regularizing $\nabla_x \mathcal{L}$ only yielded limited robustness compared to *Adversarial Training* [7], the current state of the art. We revisit this regularized training objective, and find it very effective on model architectures that use smooth activation functions, including modern vision transformer architectures [8]. In contrast to prior beliefs, our results show that more than 90% of robustness (compared to the state of the art) can be

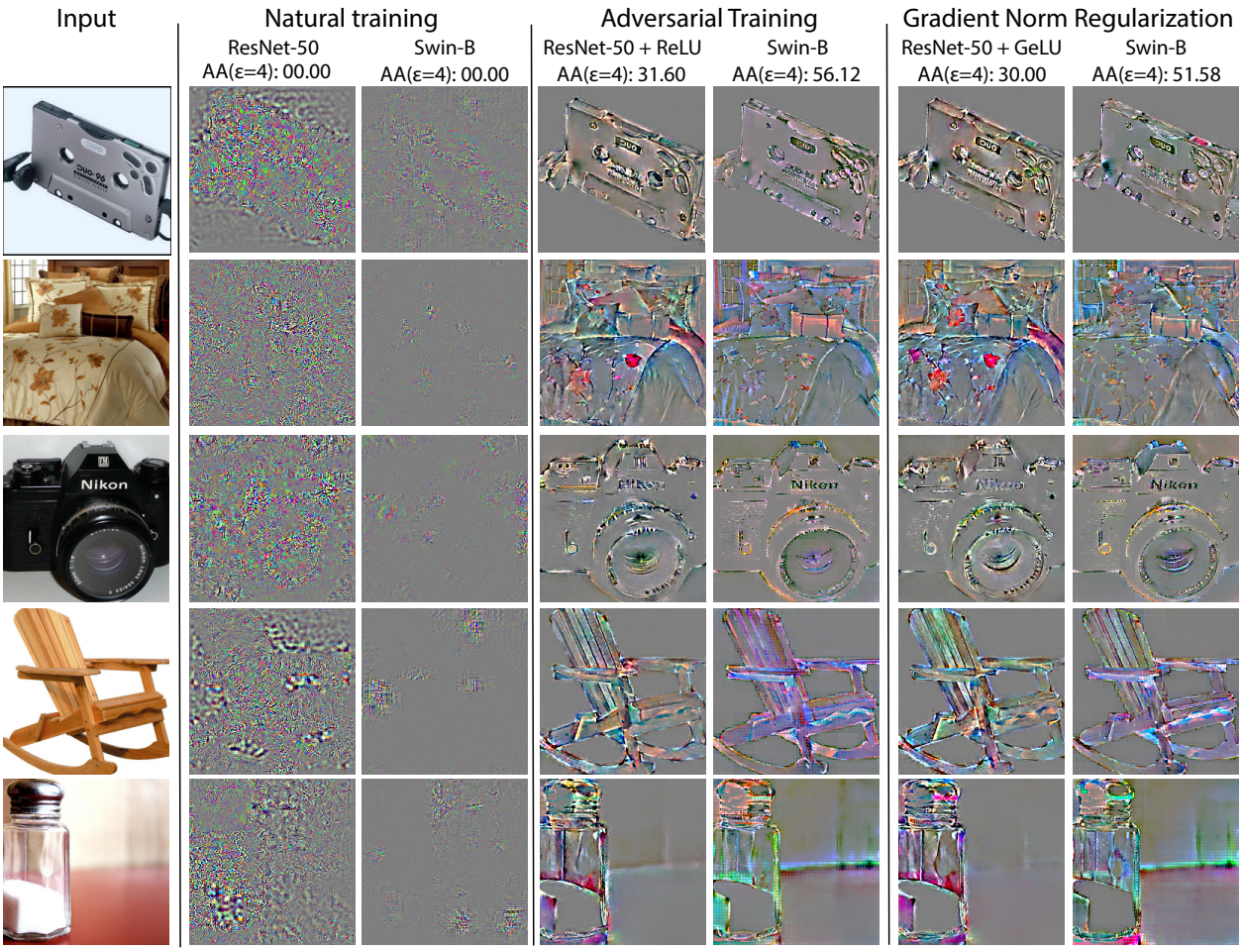

Figure 1.1: Comparison of loss-input gradients of non-robust and robust models across architectures for a set of images. Non-robust models taken from timm [3]. Adversarial training is from the work of Liu *et al*. [4]. Gradient norm regularization done with the objective in Equation (4.1) and with the same recipe as adversarial training otherwise. As can be seen, a model can be easily identified as vulnerable or robust simply by looking at clean input gradients. Gradients of robust models (adversarial training and gradient norm regularization) highly resemble the input images, and look visually similar to each other to the human eye. By contrast, gradients of vulnerable models are noise-like, bearing apparently little resemblance to each other or the input images.

obtained by simply regularizing input gradients. Alternatively, enforcing gradients to focus on image edges also yields non-trivial robustness gains. We argue that input gradients $\nabla_x \mathcal{L}$ is an important quantity in both understanding and improving robustness of deep neural networks.

In summary, our contributions are as follows:

1. Showing that close to state-of-the-art robustness on ImageNet may be achieved by regularizing the norm of *natural* input gradients

2. That the effectiveness of such approaches critically depends on the smoothness of

activation functions

3. And additionally showing how regularizing the direction of the gradients to focus on image edges, independently of the norm, can also significantly increase robustness, providing evidence towards showing that perceptual alignment induces robustness.

Chapter 2

Related works

Adversarial examples. Szegedy *et al.* [1] first identified the existence of *adversarial examples*, small perturbations imperceptible to humans but that completely fool networks. Since then, extensive researches have been conducted on the subject of adversarial examples, both on defending against such attacks [9]–[18], as well as stronger attacks [19]–[24], developing into a race between between attackers and defenders.

Training robust models. *Adversarial Training* emerged as the strongest paradigm towards training robust models in practice [2], [4], [25]–[27]. It is a complex bi-level algorithm, where at each iteration we generate a strong attack (via iterative optimization) and train the network to classify it correctly. In practice, such approaches, including the accelerated single-step Fast Gradient Sign Method (FGSM), require tuning many hyper-parameters and can be difficult to train [27]–[30]. Alternative approaches attempted to regularize models to have small input gradient norms [5]–[7], [31]–[34]. Despite strong theoretical arguments, no previous works have shown competitive performance on ImageNet from such regularizations. As far as we know, our work is the first to show its strong perfomance on modern architectures, and pinpoints activation function smoothness as the deciding factor of its effectiveness.

Perceptually aligned gradients. Previous works noted that robust models tend to have *perceptually aligned gradients*, *i.e.*, class gradients $\nabla_x f_\theta(x)_{y_t}$ that align with human perception [35]–[37] (see also Figure 1.1). In this work, we analyze the reverse implication, and ask: do perceptually aligned gradients imply robustness? The work of [38] aligned model class gradients $\nabla_x f_\theta(x)_{y_t}$ with several notions of class-representative images (defined via real images and generative models), and observed improved robustness on small datasets, but only small benefits on TinyImageNet. Our thesis shows that simply aligning gradients to image edges yields much stronger robustness gains on the more challenging full ImageNet-1k. In addition to perceptual alignment, we also highlight several other properties that separate gradients of robust and non-robust models.

Chapter 3

Experimental settings

3.1 Dataset

In this thesis, we will work with the challenging ImageNet-1K [39] dataset for all experiments. In addition to being a difficult task, the high 224x224 resolution images of ImageNet will lend us great visualization ability, improving capacity for analysis.

3.2 Architecture

We will use the current state-of-the-art architecture for Adversarial Robustness on ImageNet, Swin Transformers [8], albeit in base size. We will also start from a standard pre-trained checkpoint from timm [3] in all experiments to reduce computational expense from 300 to 100 epochs per full training run (roughly 3 weeks to 1 week reduction on 8 V100 GPUs). From the work of Liu *et al.* [4] we know the impact of using pre-trained initializations and base size models is measurable but qualitatively equivalent, so we argue it is a sufficiently strong setting for our experiments. When analysing the effect of architectural choices in Section 4.3, we will run experiments on ResNet50's [40] to allow for more extensive ablations, due to the aforementioned computational cost of training transformers.

3.3 Adversarial Training skyline and training recipe

We will use the work of Liu *et al.* [4], the current state-of-the-art for L_∞ robust accuracy on ImageNet, both as a skyline to compare performance as well as for their strong training recipes. Unless otherwise stated, as per the work of Liu *et al.* [4] all training recipes for Swin Transformers last 100 epochs, and use standard training tricks and augmentations like RandAugment [41], [42], mixup with label smoothing and random erasing [43], the AdamW optimizer with weight decay [44], and model averaging [45]–[47]. Recipes used for ResNets are equivalent but halved in length. The full recipe details for all experiments are included in the supplementary material, and we will publicly release the code.

3.4 Attack benchmark

We will use AutoAttack-L_∞ [24] with perturbation strength $\epsilon = 4$ for all main robustness evaluations following the Robustbench standard [48]. To evaluate robustness for a dense set of perturbations strengths ϵ, we will use the PGD100 attack. We use the code and test set of Liu *et al*. [4] for robustness evaluations in order to have comparable results to theirs.

Chapter 4

Regularizing gradient norm leads to robust models on modern architectures

4.1 Robust gradients have small L_1 norm

Mathematically, a function stable to perturbations of the input has a small gradient w.r.t. the input. Empirically, it has been observed that adversarial robustness (obtained with adversarial training) [11] correlates with small gradient norm [5]. As we can see in Table 4.1, this continues to hold with state-of-the-art robust transformers. The expected L_1 norm of the loss-input gradient $\nabla_x \mathcal{L} := \nabla_{\mathbf{x}} \mathcal{L}_{\text{CE}}(f_\theta(\mathbf{x}, y)_1$ is more than two orders of magnitude smaller for robust models than for their non-robust counterparts of the same architecture. Furthermore, fixing the models and taking expectations over inputs conditioning on PGD10 attack success, the gradient is much smaller when the attack fails than when it succeeds (last two columns of Table 4.1).

Despite the large amounts of theoretical and empirical evidence supporting low norms for robustness, the results of extensive previous work studying low gradient norms as a regularizer [5]–[7], [31]–[34] have so far been either inconclusive, or non-performant on large datasets like ImageNet. In the following section, we show how a simplified version of the gradient norm L_1 penalty, trained through double back-propagation [49], is within 5% of the state-of-the-art on adversarial robustness on ImageNet despite seeing only natural examples, showing how small gradients have a larger driving role in robustness than previously thought.

4.2 Regularizing for small gradient norms

Works in the literature studying gradient norm regularization have presented numerous formulations with differing details, such as optimizing Jacobian gradients [7], [31], [32], regularizing gradients of both natural and adversarial examples [7], or regularizing through a discrete scheme [33]. In this work, we study the effect of the cleanest formulation possible of the objective, as presented in equation 4 of [5] and optimized in [6], which we restate in Equation (4.1)

$$\mathcal{L}_{\text{GradNorm}}(\mathbf{x}, y) = \lambda_{\text{CE}} \mathcal{L}_{\text{CE}}(f_\theta(\mathbf{x}), y) + \lambda_{\text{GN}} \frac{\epsilon}{\sigma} \nabla_{\mathbf{x}} \mathcal{L}_{\text{CE}}(f_\theta(\mathbf{x}), y))_1 \tag{4.1}$$

Table 4.1: Accuracy, robustness, and gradient norm statistics on 10k ImageNet validation images for publicly available vulnerable and robust models from timm [3] and robustbench [48] respectively. The quantities Standard, AA, and PGD10 refer to clean accuracy, and AutoAttack and PGD10 robust accuracy respectively. The quantities $\mathbf{E}[L_1|\checkmark]$ and $\mathbf{E}[L_1|\times]$ are the conditional expectations of the loss input-gradient L_1 norm conditioned on the PGD10 attack failing and succeeding respectively.

		Accuracy			**Gradient Norm**		
Architecture	**Training**	Standard	AA	PGD10	$\mathbf{E}[L_1]$	$\mathbf{E}[L_1\|\checkmark]$	$\mathbf{E}[L_1\|\times]$
Resnet-50	Std. (He *et al.*)	76.35	00.00	00.66	273.4139	115.6848	274.4619
	Adv. (Salman *et al.*)	63.99	34.96	39.97	1.7784	0.3684	2.7172
Swin-B	Std. (Liu *et al.*)	84.84	00.00	02.77	147.8595	23.6523	151.3980
	Adv. (Liu *et al.*)	76.86	56.16	59.27	1.1708	0.3437	2.3742
Swin-L	Std. (Liu *et al.*)	86.13	00.00	02.12	113.6367	15.7218	115.7575
	Adv. (Liu *et al.*)	78.53	59.56	61.27	1.4062	0.3036	3.1503

Table 4.2: Robustness of a Swin Transformer trained with gradient norm regularization compared to natural training and state-of-the-art adversarial training on AutoAttack-L_∞. Adversarial training performed from pretrained timm [3] checkpoint using the recipe of [4].

	Clean	**AutoAttack-L_∞**		
Method	-	$\epsilon = 1$	$\epsilon = 2$	$\epsilon = 4$
Natural Training	84.19	00.00	00.00	00.00
Grad. Norm ($\lambda_{\text{CE}} = 0.8, \lambda_{\text{GN}} = 1.2$)	77.78	72.04	66.20	51.58
Adv. Train. (PGD-3, $\epsilon = 4$)	77.20	72.46	67.38	56.12

where $\mathcal{L}_{\text{CE}}$ is the cross-entropy loss, $\epsilon = 4$ is the adversarial strength, $\sigma = 0.225$ is the standard deviation used for normalization on ImageNet, and $\lambda_{\text{CE}}, \lambda_{\text{GN}}$ are weighing hyper-parameters set 0.8 and 1.2 respectively.

As we can see Table 4.2, training on the above objective yields a highly competitive model despite the costraints of seeing only natural examples and having 60% of the computational budget. On AutoAttack L_∞ with $\epsilon = 4$, the standard benchmark for ImageNet [48], gradient norm regularization obtains 51.58% robust performance compared to the 56.12% obtained by state-of-the-art adversarial training (also starting from a pretrained checkpoint) [4]. For smaller epsilons, the gap shrinks to 1.18% and 0.42% for ϵ of 2 and 1 respectively, though it may be possible that the gap would be larger if the Adversarial Training was performed with a lower ϵ than 4.

More interestingly, we also evaluate behaviour for higher values of ϵ. In Figure 4.1, we plot robust performance on PGD100 for a dense ϵ interval from 0 to 16. As we can see, while the gap to adversarial training grows larger as a function of the adversarial strength ϵ, it is always less than 10%. Additionally, for $\epsilon < 12$, accuracy is always two orders of magnitude above chance. This is an incredibly surprising result: despite gradient norm regularization working on only natural examples, without even random uniform perturbations [27], it remains strong

Table 4.3: Computational cost per batch comparison between natural training, adversarial training, and gradient norm regularization. Theoretical cost measured in number of network passes per batch, and empirical cost measured in seconds per batch. Experiments conducted on the same set of 8 V100 GPUs without mixed precision. Averages and standard deviations reported for the average batch time over three separate runs.

Method	# passes	Rel. to Adv. Train.	Empirical cost (s)	Rel to Adv. Train.
Nat. Train.	2	0.250	0.749 ± 0.00216	0.255
Grad. Norm.	5	0.625	1.848 ± 0.01108	0.628
Adv. Train. (PGD-3)	8	1.000	2.943 ± 0.00141	1.000

even for very large ϵ.

This showcases both the strong bias towards smoothness endowed by modern architectural choices, as well as the strong robustness driving effect of small gradient norms. Table 4.3 displays computational cost comparisons between natural and adversarial training compared to gradient norm regularization. Roughly speaking, it means that for smooth swin transformers, adversarial training with PGD-3 is expending 40% of its computational budget improving results by 5.1% accuracy points, or about an 8.8% relative increase. Similarly, minimizing the loss input-gradient L_1 norm is responsible for 92% of state-of-the-art robust accuracy.

In the following section, we empirically show the drastic effect of smooth non-linearities on the performance of gradient norm regularization, even on smaller architectures such as ResNets.

4.3 Smooth activation functions make gradient norm regularization effective

The formulation of the gradient norm objective in Equation (4.1) is extremely similar to that of previous works [6], [7], [32], [33], so why are results so different now? As was openly discussed by previous works [5], [33], ReLU networks are non-smooth *i.e.* they have non-differentiable gradients. While they raised concerns regarding the effect of these non-differentiable gradients on gradient norm regularization, they did not conduct tests to measure the size of this effect, which we perform in this section.

We set up the following controlled comparison. First, we take a pre-trained ResNet-50 [3] and replace all the ReLU non-linearities with smooth GeLUs and SiLUs [50]. This change causes clean accuracy to decrease to 0%, so we finetune over three epochs. For consistency, we also finetune the ResNet-50 with ReLUs with the same recipe and seed. Next, we make a copy of each network and train each copy on the same recipe (a halved version of the recipe used for transformers), with both adversarial training with PGD-3 and gradient norm regularization. Throughout all the trainings we keep the batch-normalizations in evaluation mode in order to isolate the effect of the non-linearity. The evaluations on L_∞-AutoAttack are shown on Table 4.4. We show validation accuracies as functions of epoch in Figure 4.3.

Table 4.4: Clean and L_∞-AutoAttack accuracy for ResNets with ReLU, GeLU, and SiLU non-linearities trained with both Adversarial Training and GradNorm for 50 epochs using a shortened version of the Adversarial Transformer recipe of [4].

Method		**Accuracy**	
Training	Non-linearity	Standard	AutoAttack-L_∞
GradNorm	ReLU	16.94	6.82
	GeLU	60.34	30.00
	SiLU	61.84	30.58
1-4 Adv. Train.	ReLU	59.46	31.60
	GeLU	59.34	32.64
	SiLU	60.58	33.40

As we can see in Table 4.4 and Figure 4.3, the ResNet with ReLU is completely incapable of properly fitting the objective at the appropriate strength, with clean performance sharply decaying and robust performance barely increasing, compared to the ResNet with GeLU, despite training on the same recipe with the same regularization objective and weights. In contrast, the gradient norm regularized GeLU ResNet displays similar convergence behaviour to the adversarially trained model, obtaining extremely similar final clean and robust accuracies.

The work of [51] conducted a similar analysis for Adversarial Training, observing small increases in performance from using smooth non-linearities. As we can see from Table 4.4, for Gradient Norm regularization the effect is more than 20 times larger: robust performance on AutoAttack with adversarial training increases by 1%, while for gradient norm the increase is roughly 23%. In essence, the gradient norm regularization objective minimizes a penalty on the gradients of the network; since for ReLU networks the latter is non-differentiable, Taylor's theorem does not necessarily hold on the gradient loss, so there is no guarantee that gradient descent will work.

Additionally, we found the usage of adaptive optimizers like Adam [52], as well as a relatively small warm-up learning rate of 10^{-5}, to be extremely important. Especially at the beginning, the size of the norm of the gradient changes drastically; using a non-adaptive optimizer or too high a learning rate also causes performance to similarly crash, even on smooth networks, for the regularization weights required to reach performance comparable to adversarial training. This may have been another reason behind the lack of conclusive success of previous works.

In Figure 4.2, we visualize PGD10 perturbations of the same vulnerable and robust models in Figure 1.1. Similarly as in Figure 1.1, obvious visual differences exist between the perturbations of vulnerable and robust models.

4.4 Properties of robust gradients beyond small L_1 norm

In this section, we compare and contrast gradients from adversarially trained models with those obtained through gradient norm regularization. What is present in the former and not

the latter? We briefly provide an interesting finding in that direction.

Figure 4.4 plots the distribution of the absolute values of gradients across 128 images. In Figure 4.4.a, we see that adversarially trained gradients have much smaller small values than gradient norm regularized gradients. The distribution for the later tapers off at 10^{-9}, while the former has a bump between 10^{-11} and 10^{-10}. Even more interestingly, plotting red Figure 4.4.b, green Figure 4.4.c, and blue channels Figure 4.4.d individually, we observe that the difference observed in Figure 4.4.a is almost entirely due to the green channel of the gradients. This asymmetric role of the green channel is unique to adversarial training across the three models. It's possible this may be due to the special role that the green channel plays in image coding, such as sampling using Bayer filters [53].

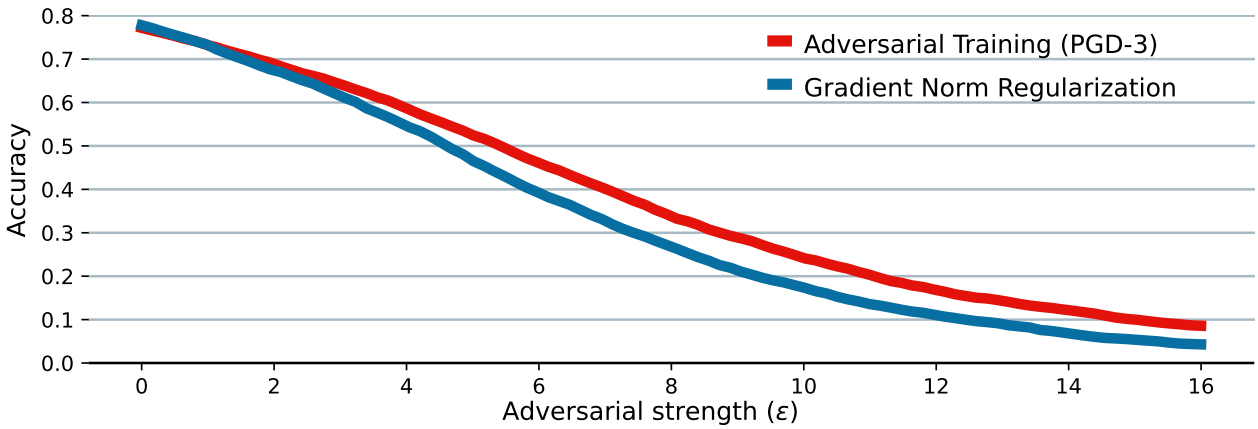

Figure 4.1: Robust accuracy vs epsilon for the PGD100 attack on ImageNet for Swin Transformer trained on Gradient Norm Regularization and state-of-the-art Adversarial Training. Gradient Norm Regularization achieves slightly better accuracy on clean images ($\epsilon = 0$) and good robust performance ($\epsilon > 0$), despite seeing only natural examples and having 60% of the computational cost of Adversarial Training with PGD-3.

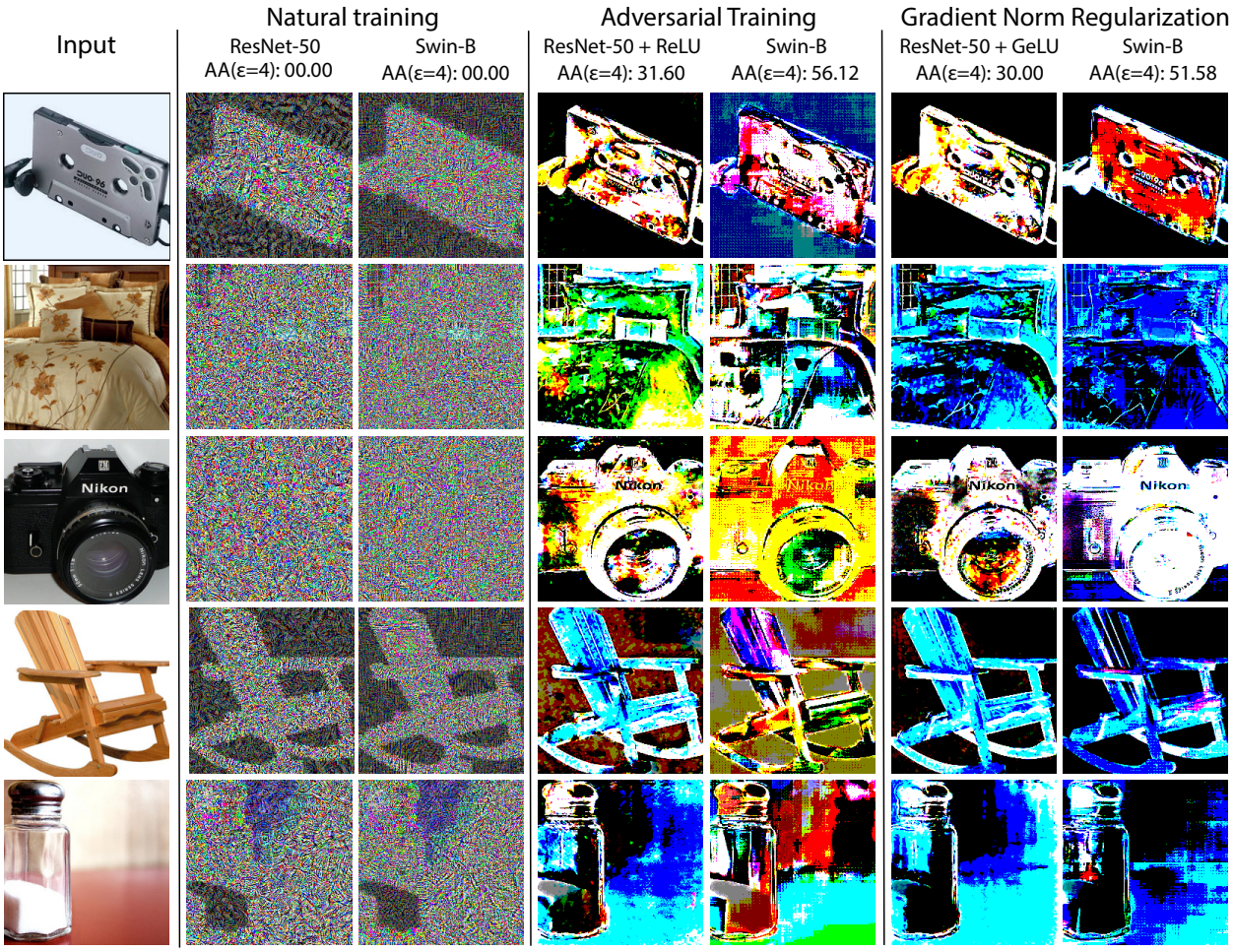

Figure 4.2: Comparison of PGD10 L_∞ (ϵ=4) perturbations of non-robust and robust models across architectures for a set of images (same as in Figure 1.1). As with clean input gradients, models can again be easily identified as vulnerable or robust simply by looking at the perturbations. Perturbations coming from robust models (adversarial training and gradient norm regularization) highly resemble the input images, though the visual similarity has decreased w.r.t. the input gradients. Perturbations originating from vulnerable models are now even more noise-like, with the exception of images with very flat backgrounds, potentially because the gradient may oscillate around zero in those areas.

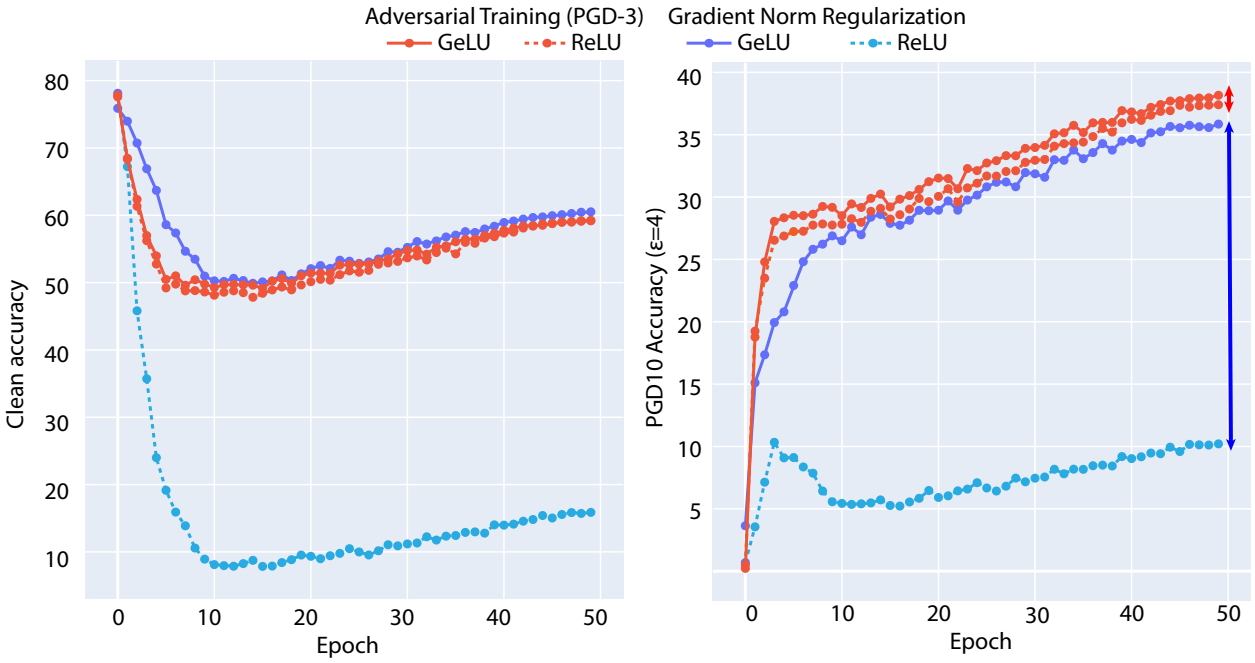

Figure 4.3: Clean and PGD10 ($\epsilon = 4$) robust accuracy vs epoch for ResNet50 with ReLU and GeLU trained with Adversarial Training and Gradient Norm Regularization. We observe how the ReLU ResNet is not capable of handling the regularization objective at the appropriate strength.

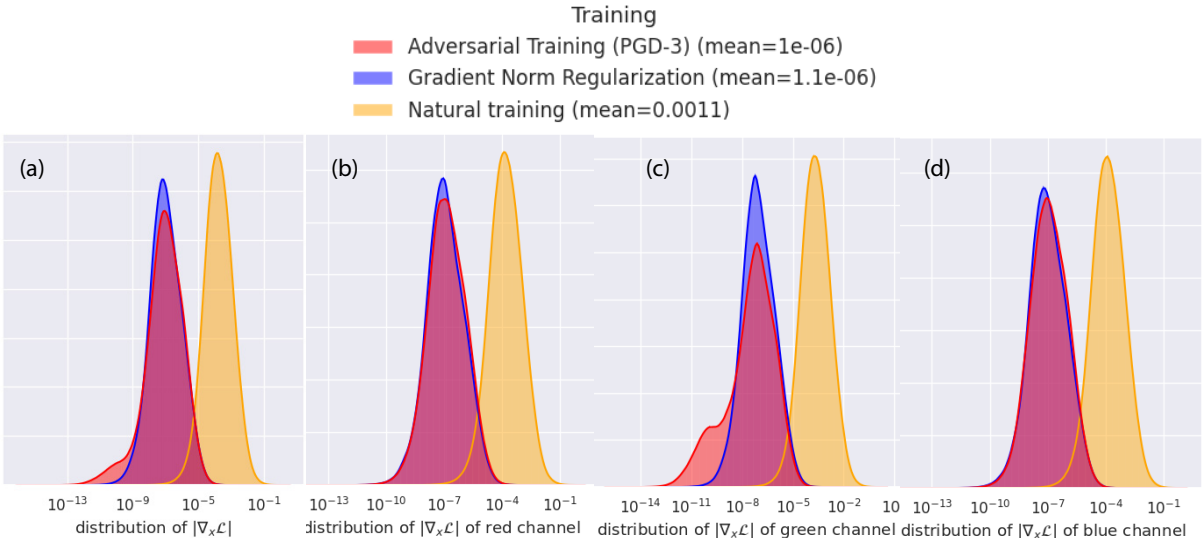

Figure 4.4: Distribution of absolute value of gradients over 128 images for (a) all channels (b) the red channel (c) the green channel and (d) the blue channel. In (a), we see adversarial gradients have a significantly fatter left tail; essentially, the small values are much lower. From looking at the channel-specifcic plots (b),(c), and (d), we observe that most of this difference is owed to the green channel: the small values of the green channel of adversarially trained gradients are very small. This asymmetric behaviour is missing from both naturally trained and gradient norm regularized models.

Chapter 5

Aligning gradient to image edges improves robustness

We also evaluate properties common to both adversarially trained and gradient norm regularized models. Despite highlighting the same object [35], [36], [54], the saliency maps (the per-pixel maximum absolute value of the class input-gradients) of vulnerable and robust models display significantly opposite correlations with the image edges, as seen visually in Figure 5.1 and numerically in Table 5.1.

The image edges, also known as the oriented energy of the image, are calculated using Sobel filters [55]; in simple terms, horizontal and vertical derivatives are calculated using a convolution, squared, and finally added to obtain a measure of the amount of local change at each pixel.

In Table 5.1, we report the log-log correlation between both the saliency map and the loss input-gradient absolute value with the oriented energy of the input. For the naturally trained model, both values are significantly negative at around -0.45, for both robust models, they are significantly positive at around +0.56.

In Figure 5.1, we visualize this similarity between saliency maps and image edges (*i.e.*, oriented energy), and show the correlation between log saliency map and log oriented energy plots for each pixel. We observe strong positive correlation for robust models, and weaker negative correlations for non-robust models.

A natural question that arises is wether this property is merely a consequence of robustness, or actually induces robust behaviour. We find that, to a signficant extent, it is actually the latter. We first tried to optimize cosine similarity between the loss-input gradient ($\nabla_x \mathcal{L}$)) and oriented energy directly. However, optimization did not converge. Instead, we obtain significant results by aligning class gradients ($\nabla_x f_\theta(x)_{y_t}$, *i.e.*, the gradient of target-class logit w.r.t. input), following [35] who regularize this quantity in a different fashion. Additionally, sampling the class according to the probabilites of the model, rather than always using the target class, marginally but measurably improved results. This final form of the regularizer is the following:

$$\mathcal{L}_{\text{edge}}(\mathbf{x}, y) \coloneqq \mathcal{L}_{\cos}(\nabla_{\mathbf{x}} f_\theta(x)_t, |g_u * x|^2 + |g_v * x|^2) \tag{5.1}$$

$$t \sim \text{softmax}(f_\theta(x)/0.5) \tag{5.2}$$

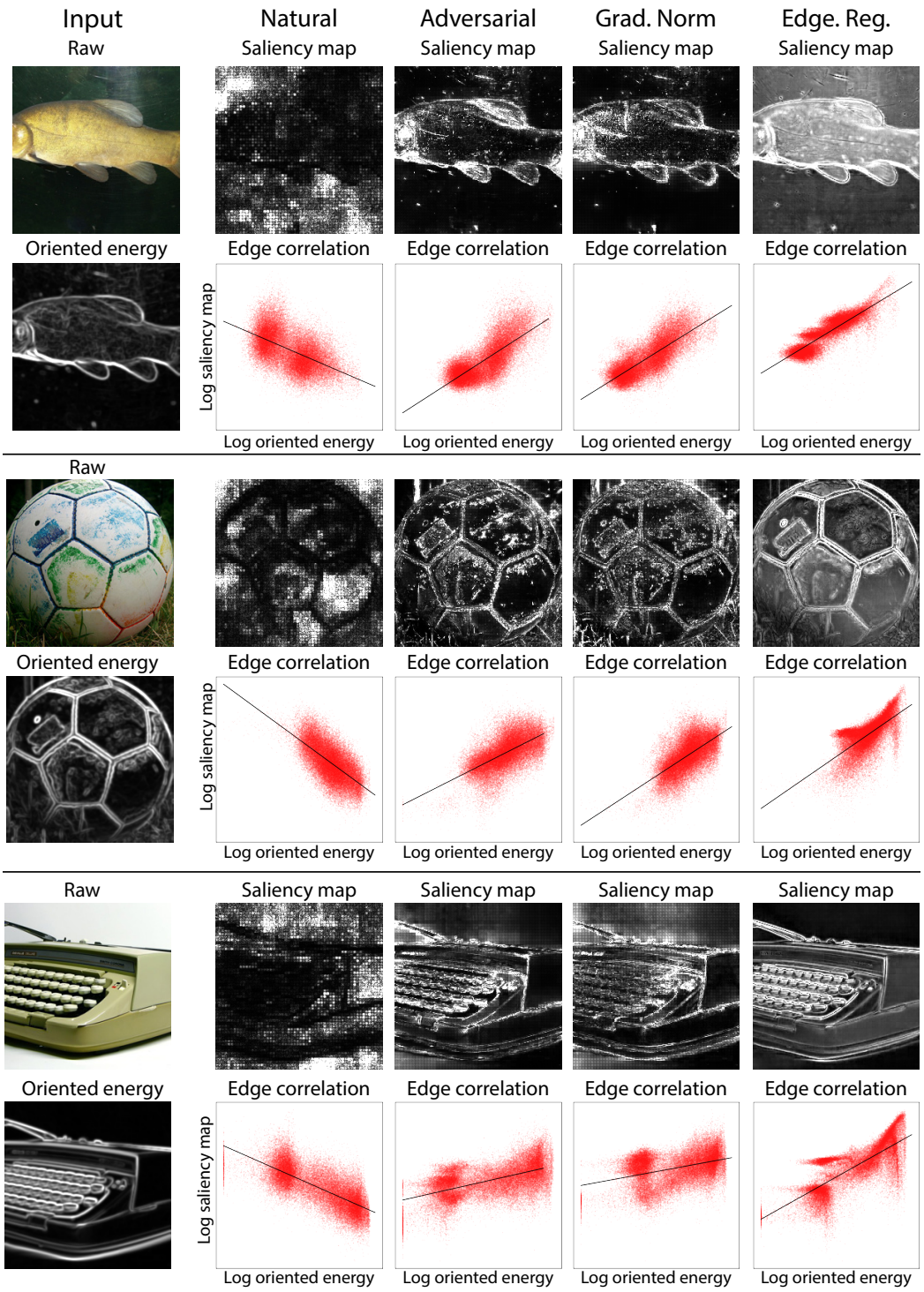

Figure 5.1: Scatter plots of log gradient magnitude vs log oriented energy for a non-robust and robust Swin Transformer. Oriented energy is calculated as $\text{edge}(x) = |g_u * x|^2 + |g_v * x|^2$, where $*$ denotes the convolution operation, and g_u, g_v denote Gaussian horizontal and vertical derivative filters respectively. Oriented energy is clamped at 1e-3 to eliminate extremely low outliers. Saliency maps are clipped to percentile 0.95 for visualization purposes only.

Table 5.1: Average and standard deviation of log saliency map magnitude vs log oriented energy Pearson correlation across 10000 validation images of ImageNet. We observe a significant positive correlation of +0.56 between the saliency map of the robust model and the oriented energy of the input, showing that the majority of the gradient content is located at the edges of the image. By contrast, the significant negative correlation of −0.45 between the saliency maps of vulnerable vanilla models and the oriented energy of the input show that the majority of the gradient is located at the flat regions of the image. Moreover, we believe these values undersell the relationship as the edges are naively calculated and highlight irrelevant objects that will have zero gradient content. Oriented energy calculated as $\text{edge}(x) = |g_u * x|^2 + |g_v * x|^2$, where $*$ denotes the convolution operation, and g_u, g_v denote Gaussian horizontal and vertical derivative filters respectively.

	Accuracy		Pearson corr. of log w/ log oriented energy	
Training	Clean	AutoAttack	Saliency map	$\lvert\nabla_x \mathcal{L}_{\text{CE}}(f_\theta(x), y)\rvert$
Natural	**84.19**	00.00	$-0.4510 \pm (0.1467)$	$-0.4428 \pm (0.1498)$
Gradient Norm	77.78	51.58	$+0.5627 \pm (0.1640)$	$+0.5679 \pm (0.1647)$
Adversarial (PGD-3)	77.20	**56.16**	$+0.5627 \pm (0.1569)$	$+\mathbf{0.5692} \pm (\mathbf{0.1569})$
1-5 Edge regularization	76.80	35.02	$+\mathbf{0.6055} \pm (\mathbf{0.1644})$	$+0.3882 \pm (0.2020)$

where $\mathcal{L}_{\cos}$ is the cosine similarity loss, $\nabla_{\mathbf{x}} f_\theta(x)_y$ is the gradient of the label logit with respect to the input, $\text{softmax}(f_\theta(x)/0.5)$ is the probability distribution defined by the model outputs with a temperature of 0.5, $*$ denotes 2D convolution, and g_u, g_v denote gaussian horizontal and vertical derivative filters respectively. Note that unlike gradient norm regularization, Equation (5.1) does nothing to a-priori regularize the norm of the gradients, only the direction.

Optimizing the above quantity turns out to be slightly more difficult, requiring 128 epochs to converge. However, as we can see in row 4 of Table 5.1, this *edge regularization* is sufficient to obtain significant robustness on AutoAttack L_∞, obtaining 35.02% robust accuracy, a relative performance of 60% compared to the current state of the art via Adversarial Training [4]. While not at the level of models regularized purposefully for robustness, this serves as significant evidence to support our hypothesis. Moreover, the true value of the result may lie in the following: it is much easier to conceptualize devising an architecture that natively focuses on edges than one that has low gradient norm or is resistant to arbitrary perturbations. That is, the success of edge regularization could potentially provide a start towards a *structurally robust* architecture *i.e.* one that displays adversarial robustness even when trained normally. We leave this direction to future work.

Chapter 6

Conclusion

What properties of input gradients characterize model robustness? In our work we find that, on architectures with smooth non-linearities, cleanly minimizing the L_1 norm of the loss input gradients achieves close to state-of-the-art robust performance, despite never training on perturbed examples. This implies that (1) model robustness is significantly characterizable by behaviour on natural inputs, and (2) architectural changes like non-linearity choice may drastically change the effectiveness of alternate approaches towards robustness. We also find an additional characterization of robust gradients based on image edges, independent of norm, that achieves 60% performance of state-of-the-art. While numerically weaker than the gradient norm result, the implications are still potentially significant. Specifically, it provides a possible hint towards a *naturally robust architecture* that, through natively enforcing dependence only on the edge regions of the image, is resistant to perturbations despite being naturally trained.

6.1 Limitations

Though we indirectly evaluate the effectiveness on ResNet50, most of our main experiments including performance comparisons to the state-of-the-art are limited to swin transformers. Additionally, there is yet still a gap between Gradient Norm Regularization and Adversarial Training that we do not characterize.

6.2 Broader impact

Deep neural networks have become the gold standard for computer vision tasks, but are also extremely brittle. To this, advances in understanding of model robustness are highly important towards the adoption of deep neural networks in safety critical tasks.

Appendix A

Second-Order Analysis

The main text focuses on analyses of first-order statistics involving input-gradients. In this section, we provide additional analyses focusing on second-order behaviors, following previous work [56]–[61]. In particular:

- Appendix A.1 measures the geometry statistics introduced by Srinivas *et al.* [57].
- Appendix A.2 measures the local linearity error of Rocamora *et al.* [56].
- Appendix A.3 measures loss $\mathcal{L}$, L_1 gradient norm $|\nabla\mathcal{L}|_1$, and normalized curvature $\frac{|\nabla^2\mathcal{L}|_2}{|\nabla\mathcal{L}|_2}$ [57] as we move in the attack direction.

In contrast to previous works on smaller datasets, our analyses focus on large-scale models trained on ImageNet.

A.1 Geometry statistics

Table A.1 plots average and standard deviation of the geometry statistics introduced by Srinivas *et al.* [57]: the loss-input gradient L_2 norm $|\nabla\mathcal{L}|_2$, the loss-input hessian spectral norm $|\nabla^2\mathcal{L}|_2$, and the normalized curvature $C_\mathcal{L} \coloneqq \frac{|\nabla^2\mathcal{L}|_2}{|\nabla\mathcal{L}|_2}$.

Standard deviations are very high due to a high variability in scale across examples, which we think is a feature unique to ImageNet as this was not observed by Srinivas *et al.*, though it could also be due to the much larger transformer model. Hence, we plot the distribution of curvature across examples in log scale in Figure A.1.

As we can see, robust training leads to a decrease in normalized curvature of over three orders of magnitude w.r.t. natural training; hessian spectral norms drop by more than six, while gradient L_2 norms by around two. Between the two robust models, all geometry statistic averages are surprisingly very similar, with the normalized curvature of the adversarially trained model actually being slightly higher. However, we also observe that the standard deviation of gradient and hessian norm are significantly higher for gradient norm regularization, despite normalized curvature standard deviation being lower; this means that gradient and hessian norms vary on the same examples, such that normalized curvature remains small. Through visual examination of Figure A.1.b, we see that the curvature distribution is also

Table A.1: Geometry statistics as presented in Srinivas *et al.* [57]. Due to the high variation of the statistic scale across images standard deviations are very high, especially for the natural model. Hence, we also show the distribution of the curvature in Figure A.1. Robust training (both gradient norm regularization and adversarial training) leads to an improvement in normalized curvature of over three orders of magnitude w.r.t. to natural training. Surprisingly, amongst the two robust models adversarial training has the highest curvature, though the numbers are quite close. Statistics calculated over 3200 ImageNet validation images.

	Accuracy		Geometry		
Training	Clean	AutoAttack	$E_{\mathbf{x}}\nabla\mathcal{L}(\mathbf{x})_2$	$E_{\mathbf{x}}\nabla^2\mathcal{L}(\mathbf{x})_2$	$E_{\mathbf{x}}C_{\mathcal{L}}(\mathbf{x})$
Natural training	**84.19**	00.00	$28.7 \pm (93.8)$	$10^6 \pm (10^7)$	$10^4 \pm (10^5)$
Gradient norm regularization	77.78	51.58	$0.31 \pm (0.67)$	0.79 ± 3.25	$2.15 \pm (1.72)$
Adversarial training (PGD-3)	77.20	**56.16**	$0.31 \pm (0.44)$	0.75 ± 1.71	$2.17 \pm (1.83)$

very similar across the two robust models. A possible explanation is that the gradient norm regularized model overfits to the clean examples, and thus displays similar numbers to the adversarially trained model despite its lower performance.

A.2 Local linearity error

As defined by Rocamora *et al.* in [56], the local linearity error optimizes the linearity of the model w.r.t. uniformly distributed perturbations. We rewrite their objective in Equation (A.1)

$$\mathbf{E}_{\alpha,\eta_1,\eta_2}\left[|\alpha\mathcal{L}(x+\eta_1)+(1-\alpha)\mathcal{L}(x+\eta_2)-\mathcal{L}(x+\alpha\eta_1+(1-\alpha)\eta_2)|^2\right] \tag{A.1}$$

$$\alpha \sim U(0,1)\ \ \eta_1,\eta_2 \sim U(-\epsilon,\epsilon) \tag{A.2}$$

where $\epsilon = 4./255$.

Table A.2 reports mean and standard deviation of the local linearity error across 10000 ImageNet validation images. As we can see, the two robust models have significantly smaller local linearity errors by about two orders of magnitude compared to the naturally trained model. Within the two robust models, adversarial training has lower average local linearity error, consistent with its superior robustness.

A.3 Loss, L_1 gradient norm, and normalized curvature along attack direction

In this section, we measure how statistics change along the attack direction by linearly interpolating between a clean and attacked input. While we use the relatively weak PGD-5 attack to reduce computational expense, as we can see in the legend of Figure A.3 that it is

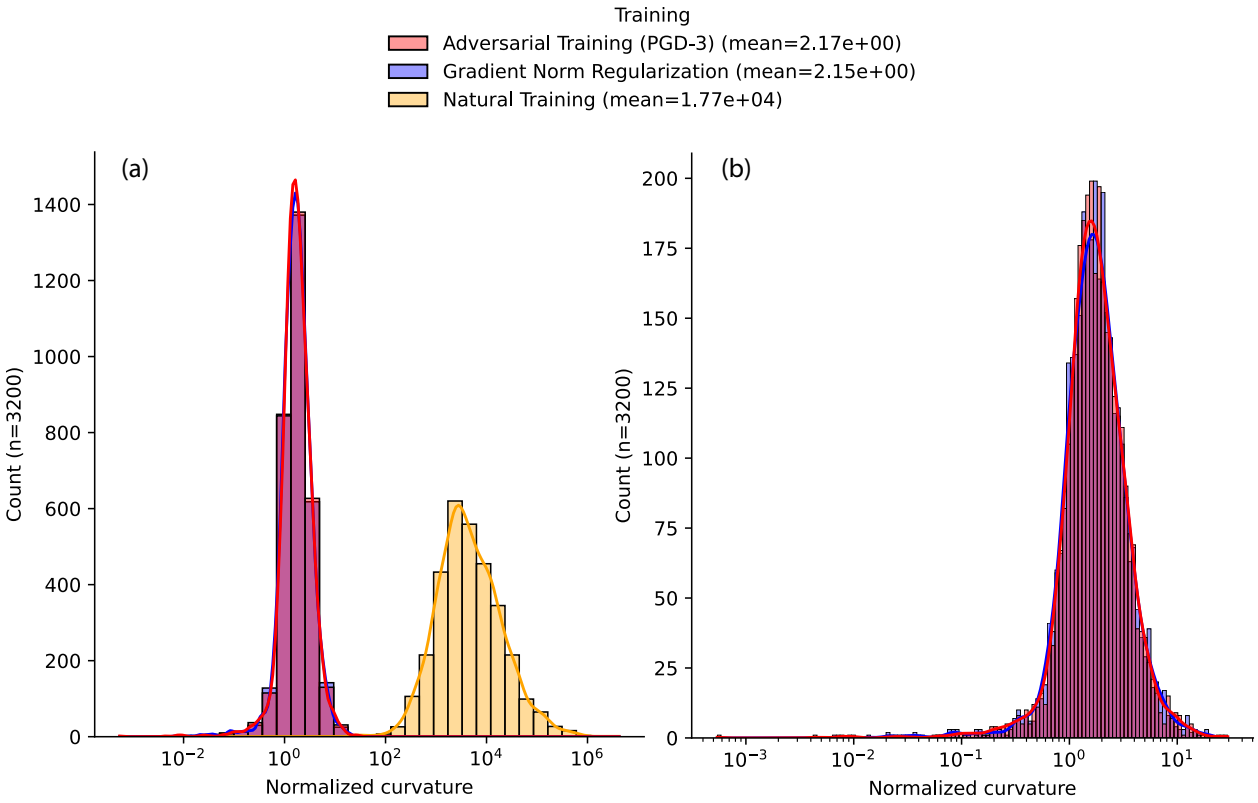

Figure A.1: Distribution of normalized curvature [57] over 3200 ImageNet validation images, (a) plots all three models and (b) plots just the robust models. Due to the high variation in scale over images, the graph is plotted with the x-axis in log scale. In (a) we see robust training (both gradient norm regularization and adversarial training) leads to an improvement in normalized curvature of over three orders of magnitude w.r.t. to natural training. Surprisingly, amongst the robust models adversarial training has the slightly higher average curvature, though the distributions are so close they cannot be distinguished in the graph.

enough to clearly separate the three models in terms of robustness. The interpolated attack is calculated as follows

$$x(\epsilon) = x + \frac{\epsilon}{4}(\text{PGD-5}_{\epsilon=4}(x) - x) \tag{A.3}$$

Figure A.3.a1 plots average classification loss as function of interpolation ϵ for the three models. As we can see, the loss of the naturally trained model quickly grows to worse-than-random (6.9). Figure A.3.a2 zooms into the two robust models, where we observe two key things: (1) adversarial training has higher loss very close to the origin but lower loss away, and (2) despite their robustness loss still significantly increases as we move in the attack direction.

Figure A.3.b1 and Figure A.3.b2 plots L_1 loss-input gradient norm as a function of interpolation ϵ in the same manner, with similar conclusions. Despite having very similar gradient norms at the origin, gradient norm regularization quickly becomes worse than adversarial training away from the origin. Moreover, both models display a high increase in gradient norm along the attack direction w.r.t. the value at the origin.

Figure A.3.c1 and Figure A.3.c2 plot normalized curvature. In this particular sample of

Table A.2: Average and standard deviation of the local linearity error of Rocamora *et al.* [56]. Due to the high variation of the scale across images, standard deviations are larger than the mean. Hence, we also report mean and standard deviation of the base 10 logarithm of the error. We add 10^{-17} before computing log statistics for numerical stability. Robust training (both gradient norm regularization and adversarial training) leads to an improvement in local linearity error of two orders of magnitude w.r.t. to natural training. Within the robust models, adversarial training has significantly lower local linearity error, reflecting its superior robustness and second-order stability. Statistics calculated for 10000 ImageNet validation images.

	Accuracy		Local linearity error	
Training	Clean	AutoAttack	linerr	$\log_{10}$(linerr)
Natural Training	**84.19**	00.00	$3.55e-3 \pm (2.69e-2)$	$-4.57 \pm (1.63)$
Gradient norm regularization	77.78	51.58	$1.86e-5 \pm (1.65e-4)$	$-6.60 \pm (1.43)$
Adversarial training (PGD-3)	77.20	**56.16**	$\mathbf{8.2e-6 \pm (5.08e-5)}$	$\mathbf{-6.84 \pm (1.48)}$

1000 images (and using less iterations when calculating hessian spectral norm), normalized curvature at the origin is slightly higher for gradient norm regularization, though they are still quite similar. However, as we step away from the origin in the attack direction, curvature for gradient norm regularization spikes relative to adversarial training. This is consistent with the relative increase in gradient norm w.r.t. the adversarially trained model seen in Figure A.3.b2.

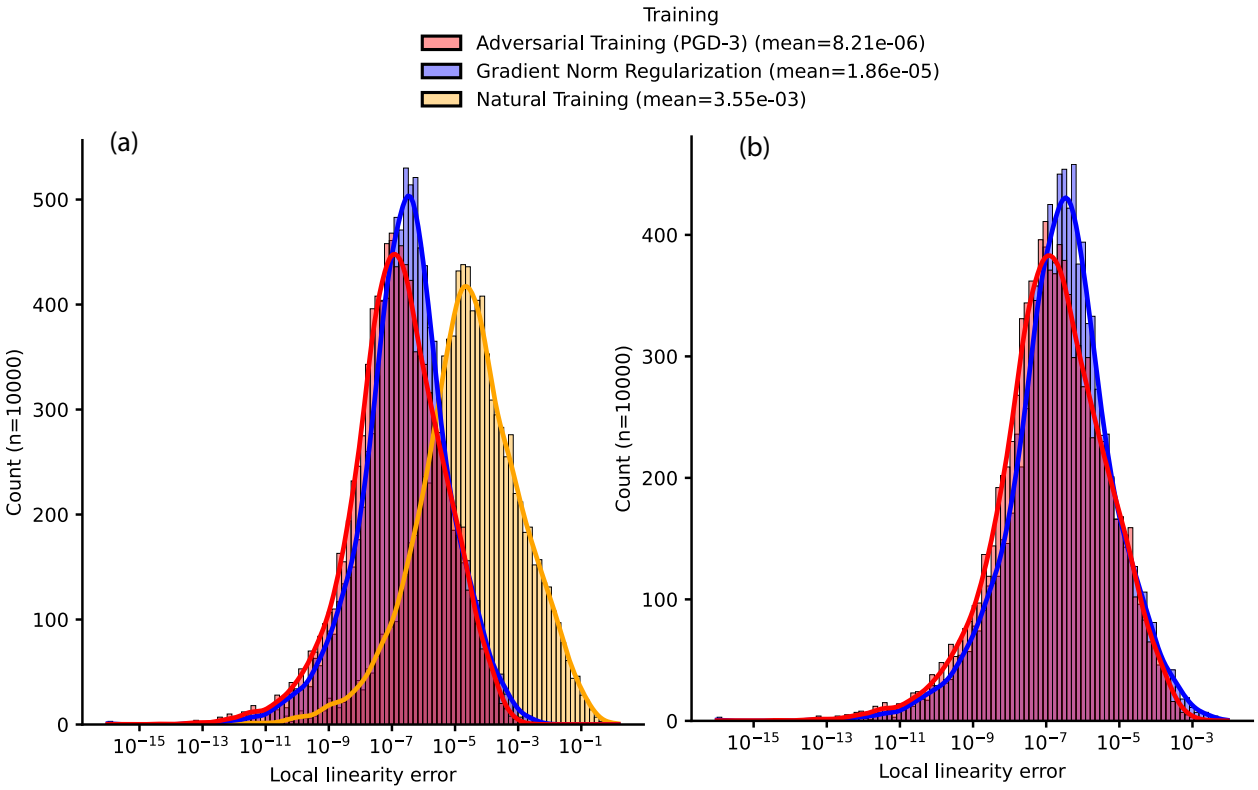

Figure A.2: Distribution of local linearity error [56] over 10000 ImageNet validation images, (a) plots all three models and (b) plots just the robust models. Due to the high variation in scale over images, the graph is plotted with the x-axis in log scale. In (a) we see robust training (both gradient norm regularization and adversarial training) leads to a visible leftward shift of the error distribution by about two orders of magnitude w.r.t. natural training. Within the robust models, adversarial training has the lowest local linearity error. The error distribution of adversarial training is shifted slightly to the left and has a much lower peak, meaning that it has a greater number of images with very low local linearity error.

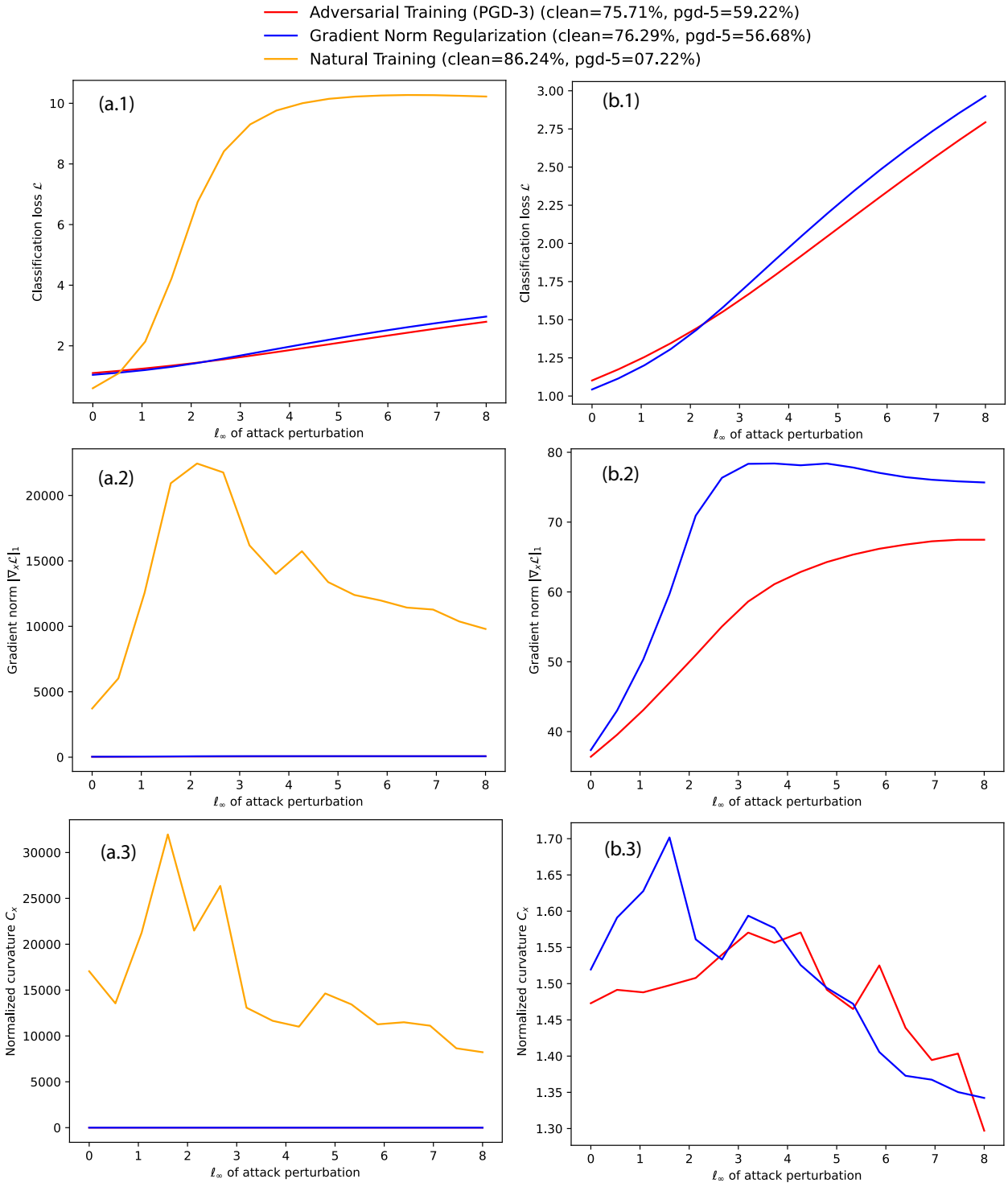

Figure A.3: Average loss (top), L_1 gradient norm (middle), and normalized curvature (bottom) [57] along the PGD-5 ($\epsilon = 4$) attack direction. The left column (a) shows all three models, where we see the extreme brittleness of natural training. The right column (b) zooms into the two robust models. Despite their robustness, loss and gradient norm significantly increase along the attack direction for both models. Comparing the two, L_1 gradient norm and curvature at the origin are similar, but gradient norm regularization has significantly higher curvature slightly away from the origin, resulting in higher losses and gradient norms along the attack direction. Statistics calculated on 1000 ImageNet validation examples. Power iteration for the normalized curvature done with 2 iterations and initialized from gradients in order to reduce computational cost.

Appendix B

Training details

We used the code of Liu *et al.* [4] as the basis for our experiments. The training configuration used in our experiments is found below in yaml format.

B.1 ResNet50

B.2 Adversarial training (PGD-3)

Same as the transformer recipe of Liu *et al.* [4], but shortened to 50 epochs. Training done in evaluation mode.

```
# optimizer parameters
opt: adamw
opt_eps: 1.0e-8
opt_betas: null
momentum: 0.9
weight_decay: 0.05
clip_grad: null
clip_mode: norm
layer_decay: null

# lr schedule
epochs: 50
sched: cosine
lrb: 1.25e-3
warmup_lr: 1.0e-6
min_lr: 1.0e-5
epoch_repeats: 0
start_epoch: null
decay_epochs: 15
warmup_epochs: 5
cooldown_epochs: 0
```

```
patience_epochs: 0
decay_rate: 0.1

# dataset parameters
batch_size: 256

# augmentation
no_aug: False
color_jitter: 0.4
aa: rand-m9-mstd0.5-inc1
aug_repeats: 0
aug_splits: 0
jsd_loss: False
# random erase
reprob: 0.25
remode: pixel
recount: 1
resplit: False
mixup: 0.8
cutmix: 1.0
cutmix_minmax: null
mixup_prob: 1.0
mixup_switch_prob: 0.5
mixup_mode: batch
mixup_off_epoch: 0
smoothing: 0.1
train_interpolation: bicubic
# drop connection
drop: 0.0
drop_path: 0.0
drop_block: null

# ema
model_ema: True
model_ema_force_cpu: False
model_ema_decay: 0.9998

# adversarial training
attack_eps: 0.01568627450980392 # 4./255.
attack_it: 3
attack_step: 0.01045751633986928 # 8./255./3.
```

Additionally, the attack step is warmed up linearly over 5 epochs as per Liu *et al.* [4].

B.2.1 Gradient Norm Regularization

Same as above, but changing adversarial training for gradient norm regularization with the following weights.

```
# gradient norm regularization
ce_weight: 0.5
gradnorm_weight: 0.5
```

Additionally, `gradnorm_weight` is warmed up linearly over 5 epochs.

B.3 Swin Transformers

B.3.1 Adversarial Training (PGD-3)

Exactly the recipe of Liu *et al*. [4].

```
# optimizer parameters
opt: adamw
opt_eps: 1.0e-8
opt_betas: null
momentum: 0.9
weight_decay: 0.05
clip_grad: null
clip_mode: norm
layer_decay: null

# lr schedule
epochs: 100
sched: cosine
lrb: 1.25e-3
warmup_lr: 1.0e-6
min_lr: 1.0e-5
epoch_repeats: 0
start_epoch: null
decay_epochs: 30
warmup_epochs: 5
cooldown_epochs: 0
patience_epochs: 0
decay_rate: 0.1

# dataset parameters
batch_size: 512

# augmentation
```

```
no_aug: False
color_jitter: 0.4
aa: rand-m9-mstd0.5-inc1
aug_repeats: 0
aug_splits: 0
jsd_loss: False
# random erase
reprob: 0.25
remode: pixel
recount: 1
resplit: False
mixup: 0.8
cutmix: 1.0
cutmix_minmax: null
mixup_prob: 1.0
mixup_switch_prob: 0.5
mixup_mode: batch
mixup_off_epoch: 0
smoothing: 0.1
train_interpolation: bicubic
# drop connection
drop: 0.0
drop_path: 0.0
drop_block: null

# ema
model_ema: True
model_ema_force_cpu: False
model_ema_decay: 0.9998
```

Additionally, the attack step is warmed up linearly over 5 epochs as per Liu *et al.* [4].

B.3.2 Gradient Norm Regularization

Same as above, except for changing adversarial training for gradient norm regularization (see Equation (4.1)) with weights ($\lambda_{\text{CE}} = 0.8, \lambda_{\text{GN}} = 1.2$), a halving of the learning rate to $0.625e-3$, an increase in the batch size from 512 to 532, and an increase in the warm-up learning rate from $1e-6$ to $1e-5$.

```
# lr schedule
lrb: 0.625e-3
warmup_lr: 1.0e-5

# dataset parameters
batch_size: 532
```

```
# gradient norm regularization
ce_weight: 0.8
gradnorm_weight: 1.2
```

Additionally, `gradnorm_weight` is warmed up linearly over 5 epochs.

References

[1] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, *Intriguing properties of neural networks*, arXiv:1312.6199 [cs], Feb. 2014. DOI: 10.48550/arXiv.1312.6199. URL: http://arxiv.org/abs/1312.6199 (visited on 03/03/2024).

[2] A. Athalye, N. Carlini, and D. Wagner, *Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples*, arXiv:1802.00420 [cs], Jul. 2018. URL: http://arxiv.org/abs/1802.00420 (visited on 02/26/2024).

[3] R. Wightman, *PyTorch Image Models*, Publication Title: GitHub repository, 2019. DOI: 10.5281/zenodo.4414861. URL: https://github.com/rwightman/pytorch-image-models.

[4] C. Liu, Y. Dong, W. Xiang, X. Yang, H. Su, J. Zhu, Y. Chen, Y. He, H. Xue, and S. Zheng, *A Comprehensive Study on Robustness of Image Classification Models: Benchmarking and Rethinking*, arXiv:2302.14301 [cs], Feb. 2023. DOI: 10.48550/arXiv.2302.14301. URL: http://arxiv.org/abs/2302.14301 (visited on 02/26/2024).

[5] C.-J. Simon-Gabriel, Y. Ollivier, L. Bottou, B. Schölkopf, and D. Lopez-Paz, *First-order Adversarial Vulnerability of Neural Networks and Input Dimension*, arXiv:1802.01421 [cs, stat], Jun. 2019. URL: http://arxiv.org/abs/1802.01421 (visited on 02/26/2024).

[6] A. S. Ross and F. Doshi-Velez, *Improving the Adversarial Robustness and Interpretability of Deep Neural Networks by Regularizing their Input Gradients*, arXiv:1711.09404 [cs], Nov. 2017. URL: http://arxiv.org/abs/1711.09404 (visited on 02/26/2024).

[7] I. Seck, G. Loosli, and S. Canu, *L1-norm double backpropagation adversarial defense*, arXiv:1903.01715 [cs], Mar. 2019. DOI: 10.48550/arXiv.1903.01715. URL: http://arxiv.org/abs/1903.01715 (visited on 02/26/2024).

[8] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 012–10 022.

[9] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial Machine Learning at Scale," in *International Conference on Learning Representations*, 2017.

[10] F. Tramèr, A. Kurakin, N. Papernot, D. Boneh, and P. McDaniel, "Ensemble Adversarial Training: Attacks and Defenses," in *International Conference on Learning Representations*, 2018.

[11] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards Deep Learning Models Resistant to Adversarial Attacks," in *International Conference on Learning Representations*, 2018.

[12] C. Guo, M. Rana, M. Cisse, and L. V. D. Maaten, "Countering Adversarial Images using Input Transformations," in *International Conference on Learning Representations*, 2018.

[13] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. Yuille, "Mitigating Adversarial Effects Through Randomization," in *International Conference on Learning Representations*, 2018.

[14] E. Wong and Z. Kolter, "Provable defenses against adversarial examples via the convex outer adversarial polytope," in *International Conference on Machine Learning*, 2018, pp. 5286–5295.

[15] F. Liao, M. Liang, Y. Dong, T. Pang, X. Hu, and J. Zhu, "Defense against adversarial attacks using high-level representation guided denoiser," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1778–1787.

[16] J. M. Cohen, E. Rosenfeld, and J. Z. Kolter, "Certified adversarial robustness via randomized smoothing," in *International Conference on Machine Learning*, 2019, pp. 1310–1320.

[17] Z. Zhang, C. Jung, and X. Liang, "Adversarial defense by suppressing high-frequency components," *arXiv preprint arXiv:1908.06566*, 2019.

[18] T. Pang, X. Yang, Y. Dong, H. Su, and J. Zhu, *Bag of Tricks for Adversarial Training*, arXiv:2010.00467 [cs, stat], Mar. 2021. URL: http://arxiv.org/abs/2010.00467 (visited on 02/22/2024).

[19] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *International Conference on Learning Representations*, 2015.

[20] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *arXiv preprint arXiv:1607.02533*, 2016.

[21] N. Carlini, A. Athalye, N. Papernot, W. Brendel, J. Rauber, D. Tsipras, I. Goodfellow, and A. Madry, "On evaluating adversarial robustness," *arXiv preprint arXiv:1902.06705*, 2019.

[22] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9185–9193.

[23] C. Xie, Z. Zhang, Y. Zhou, S. Bai, J. Wang, Z. Ren, and A. L. Yuille, "Improving transferability of adversarial examples with input diversity," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2730–2739.

[24] F. Croce and M. Hein, "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks," in *International Conference on Machine Learning*, 2020, pp. 2206–2216.

[25] Y. Dong, Q.-A. Fu, X. Yang, T. Pang, H. Su, Z. Xiao, and J. Zhu, "Benchmarking adversarial robustness on image classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 321–331.

[26] H. Salman, A. Ilyas, L. Engstrom, A. Kapoor, and A. Madry, "Do adversarially robust imagenet models transfer better?" In *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 3533–3545.

[27] E. Wong, L. Rice, and J. Z. Kolter, *Fast is better than free: Revisiting adversarial training*, arXiv:2001.03994 [cs, stat], Jan. 2020. URL: http://arxiv.org/abs/2001.03994 (visited on 02/26/2024).

[28] Z. Li, L. Liu, Z. Wang, Y. Zhou, and C. Xie, *Bag of Tricks for FGSM Adversarial Training*, arXiv:2209.02684 [cs], Sep. 2022. URL: http://arxiv.org/abs/2209.02684 (visited on 02/22/2024).

[29] M. Andriushchenko and N. Flammarion, *Understanding and Improving Fast Adversarial Training*, arXiv:2007.02617 [cs, stat], Oct. 2020. URL: http://arxiv.org/abs/2007.02617 (visited on 02/22/2024).

[30] H. Kim, W. Lee, and J. Lee, *Understanding Catastrophic Overfitting in Single-step Adversarial Training*, arXiv:2010.01799 [cs, eess, stat], Dec. 2020. DOI: 10.48550/arXiv.2010.01799. URL: http://arxiv.org/abs/2010.01799 (visited on 03/02/2024).

[31] D. Liu, L. Wu, H. Zhao, F. Boussaid, M. Bennamoun, and X. Xie, *Jacobian Norm with Selective Input Gradient Regularization for Improved and Interpretable Adversarial Defense*, arXiv:2207.13036 [cs], Nov. 2022. URL: http://arxiv.org/abs/2207.13036 (visited on 02/26/2024).

[32] D. Jakubovitz and R. Giryes, *Improving DNN Robustness to Adversarial Attacks using Jacobian Regularization*, arXiv:1803.08680 [cs, stat], May 2019. URL: http://arxiv.org/abs/1803.08680 (visited on 02/26/2024).

[33] C. Finlay and A. M. Oberman, *Scaleable input gradient regularization for adversarial robustness*, arXiv:1905.11468 [cs, stat], Oct. 2019. URL: http://arxiv.org/abs/1905.11468 (visited on 02/26/2024).

[34] C. Lyu, K. Huang, and H.-N. Liang, *A Unified Gradient Regularization Family for Adversarial Examples*, arXiv:1511.06385 [cs, stat], Nov. 2015. URL: http://arxiv.org/abs/1511.06385 (visited on 02/26/2024).

[35] S. Kaur, J. Cohen, and Z. C. Lipton, *Are Perceptually-Aligned Gradients a General Property of Robust Classifiers?* arXiv:1910.08640 [cs, stat], Oct. 2019. URL: http://arxiv.org/abs/1910.08640 (visited on 02/26/2024).

[36] S. Santurkar, D. Tsipras, B. Tran, A. Ilyas, L. Engstrom, and A. Madry, *Image Synthesis with a Single (Robust) Classifier*, arXiv:1906.09453 [cs, stat], Aug. 2019. DOI: 10.48550/arXiv.1906.09453. URL: http://arxiv.org/abs/1906.09453 (visited on 02/22/2024).

[37] S. Srinivas, S. Bordt, and H. Lakkaraju, *Which Models have Perceptually-Aligned Gradients? An Explanation via Off-Manifold Robustness*, arXiv:2305.19101 [cs], May 2023. URL: http://arxiv.org/abs/2305.19101 (visited on 02/26/2024).

[38] R. Ganz, B. Kawar, and M. Elad, *Do Perceptually Aligned Gradients Imply Adversarial Robustness?* arXiv:2207.11378 [cs], Aug. 2023. URL: http://arxiv.org/abs/2207.11378 (visited on 02/26/2024).

[39] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[40] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[41] E. D. Cubuk, B. Zoph, J. Shlens, and Q. Le, "RandAugment: Practical Automated Data Augmentation with a Reduced Search Space," in *Advances in Neural Information Processing Systems*, 2020, pp. 18 613–18 624.

[42] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation policies from data," *arXiv preprint arXiv:1805.09501*, 2018.

[43] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond Empirical Risk Minimization," in *International Conference on Learning Representations*, 2018.

[44] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[45] P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. G. Wilson, "Averaging weights leads to wider optima and better generalization," in *34th Conference on Uncertainty in Artificial Intelligence*, 2018, pp. 876–885.

[46] S. Gowal, C. Qin, J. Uesato, T. Mann, and P. Kohli, "Uncovering the limits of adversarial training against norm-bounded adversarial examples," *arXiv preprint arXiv:2010.03593*, 2020.

[47] D. S. Bolme, B. A. Draper, and J. R. Beveridge, "Average of synthetic exact filters," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 2105–2112.

[48] F. Croce, M. Andriushchenko, V. Sehwag, E. Debenedetti, N. Flammarion, M. Chiang, P. Mittal, and M. Hein, "RobustBench: A standardized adversarial robustness benchmark," in *Neural Information Processing Systems Datasets and Benchmarks Track*, 2021.

[49] H. Drucker and Y. Le Cun, "Improving generalization performance using double back-propagation," *IEEE Transactions on Neural Networks*, vol. 3, no. 6, pp. 991–997, Nov. 1992, Conference Name: IEEE Transactions on Neural Networks, ISSN: 1941-0093. DOI: 10.1109/72.165600. URL: https://ieeexplore.ieee.org/document/165600/authors#authors (visited on 02/26/2024).

[50] D. Hendrycks and K. Gimpel, *Gaussian Error Linear Units (GELUs)*, arXiv:1606.08415 [cs], Jun. 2023. DOI: 10.48550/arXiv.1606.08415. URL: http://arxiv.org/abs/1606.08415 (visited on 03/06/2024).

[51] C. Xie, M. Tan, B. Gong, A. Yuille, and Q. V. Le, *Smooth Adversarial Training*, arXiv:2006.14536 [cs], Jul. 2021. URL: http://arxiv.org/abs/2006.14536 (visited on 02/22/2024).

[52] D. P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*, arXiv:1412.6980 [cs], Jan. 2017. DOI: 10.48550/arXiv.1412.6980. URL: http://arxiv.org/abs/1412.6980 (visited on 03/07/2024).

[53] R. Lukac and K. Plataniotis, "Color filter arrays: Design and performance analysis," *IEEE Transactions on Consumer Electronics*, vol. 51, no. 4, pp. 1260–1267, Nov. 2005, Conference Name: IEEE Transactions on Consumer Electronics, ISSN: 1558-4127. DOI: 10.1109/TCE.2005.1561853. URL: https://ieeexplore.ieee.org/document/1561853 (visited on 03/07/2024).

[54] K. Simonyan, A. Vedaldi, and A. Zisserman, *Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps*, arXiv:1312.6034 [cs], Apr. 2014. URL: http://arxiv.org/abs/1312.6034 (visited on 02/26/2024).

[55] W. Freeman and E. Adelson, "The design and use of steerable filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 9, pp. 891–906, Sep. 1991, Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence, ISSN: 1939-3539. DOI: 10.1109/34.93808. URL: https://ieeexplore.ieee.org/document/93808 (visited on 03/03/2024).

[56] E. A. Rocamora, F. Liu, G. Chrysos, P. M. Olmos, and V. Cevher, "Efficient local linearity regularization to overcome catastrophic overfitting," en, Oct. 2023. URL: https://openreview.net/forum?id=SZzQz8ikwg (visited on 02/22/2024).

[57] S. Srinivas, H. Lakkaraju, K. Matoba, and F. Fleuret, "Efficient Training of Low-Curvature Neural Networks," en,

[58] S.-M. Moosavi-Dezfooli, A. Fawzi, J. Uesato, and P. Frossard, *Robustness via curvature regularization, and vice versa*, arXiv:1811.09716 [cs, stat], Nov. 2018. DOI: 10.48550/arXiv.1811.09716. URL: http://arxiv.org/abs/1811.09716 (visited on 03/12/2024).

[59] T. Tsiligkaridis and J. Roberts, *Second Order Optimization for Adversarial Robustness and Interpretability*, arXiv:2009.04923 [cs, stat], Sep. 2020. URL: http://arxiv.org/abs/2009.04923 (visited on 03/14/2024).

[60] C. Qin, J. Martens, S. Gowal, D. Krishnan, K. Dvijotham, A. Fawzi, S. De, R. Stanforth, and P. Kohli, "Adversarial Robustness through Local Linearization," in *Advances in Neural Information Processing Systems*, vol. 32, Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper_files/paper/2019/hash/0defd533d51ed0a10c5c9dbf93ee78a5-Abstract.html (visited on 03/14/2024).

[61] V. Singla, S. Singla, D. Jacobs, and S. Feizi, *Low Curvature Activations Reduce Overfitting in Adversarial Training*, arXiv:2102.07861 [cs], Aug. 2021. URL: http://arxiv.org/abs/2102.07861 (visited on 02/22/2024).