Improving Adversarial Robustness of Vision Transformers

Anonymous Author(s)
Affiliation
Address
email

Abstract

Vision Transformers (ViTs) have recently achieved competitive performance in broad vision tasks. Unfortunately, on popular threat models, naturally trained ViTs are proven to provide no more adversarial robustness than convolutional neural networks (CNNs). Adversarial training is still required by ViTs to defend against such adversarial attacks. This paper provides a comprehensive evaluation of various training techniques across several datasets, thus bringing the first implementation benchmark for adversarial training of ViTs. In addition, regarding ViT as a new type of model architecture, we further investigate its adversarial robustness from the perspective of its unique architectural components. We find, when we randomly mask the perturbation on some of the patches or the gradient from some attention blocks during adversarial training, the adversarial robustness of ViTs can be further improved. Our work may potentially open up a line of work to explore the architectural information inside new models like ViTs.

1 Introduction

Recent years have witnessed the overwhelming advances of Vision Transformers (ViTs) [1, 2, 3]. Different from widely deployed Convolutional Neural Networks (CNNs) [4, 5, 6, 7] that adopt a series of local convolutional operations to input images, ViTs adopt self-attention mechanisms [8] to sequences of image patches. Based on large scale pre-training, ViTs have achieved competitive and even better performance compared to CNNs in several fields such as semantic segmentation [9], object detection [10], and image generation [11].

Despite their great success in a growing number of vision tasks, ViTs fail in providing more adversarial robustness than CNNs on popular threat models [12]. To defend against such adversarial examples crafted by adding human-imperceptible perturbations to images [13, 14], ViTs still demand adversarial training (AT) [15, 16, 17, 18], an approach that incorporates adversarial examples into training and obtains notable empirical robustness. Note that implementation details make a big difference in AT [19], which motivates us to comprehensively evaluate various training techniques on benchmark datasets, and thus provide the first implementation benchmark for AT of ViTs. We find that gradient clipping and pre-training on large datasets are necessary for improving AT of ViTs, and advanced data augmentations indeed improve robustness in contrary to CNNs. SGD behaves better for ViTs than AdamW in term of adversarial robustness. Robust overfitting occurs much easier during AT of ViTs. Based on our empirical findings, we provide a practical recipe for adversarial training of ViTs to help researchers in their future work.

Although ViTs can achieve reasonable adversarial robustness via AT just like CNNs, it is unexplored whether or not we are able to further improve the robustness based on the specific architectural information from ViTs, a model totally different from CNNs. Interestingly, when we randomly mask the perturbation on some of the patches or the gradient from some attention blocks during adversarial

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

training, the robustness of ViTs can be further improved. We term these two simple methods as *Perturbation Random Masking* (PRM) and *Attention Random Dropping* (ARD) respectively, and conduct extensive experiments to verify their effectiveness. Our main contributions are summarized as follows:

- We comprehensively evaluate training techniques for AT of ViTs on popular threat models, which indicates the necessity of gradient clipping and pre-training, proper choices of advanced data augmentations and optimizers, and the higher propensity to robust overfitting compared to CNNs.
- Based on the unique architectural information from ViTs, we propose *Perturbation Random Masking* (PRM) and *Attention Random Dropping* (ARD) as a warming-up strategy to improve the adversarial robustness. In particular, PRM randomly masks the perturbation on some of the patches, while ARD randomly masks the gradient from some attention blocks.
- Our work not only provides the first implementation benchmark for AT of ViTs, but also reminds researchers of the potential of architectural information inside brand new models like ViTs.

2 Related Work

2.1 Training Strategies for ViTs

Due to the lack of inductive bias [1], ViTs require more training techniques to reach their performance potential. Dosovitskiy et al [1] first pretrained ViTs on the large-scale datasets (ImageNet-21k or JFT-300M [20]) and achieved competitive performance on downstream tasks. Touvron et al [3] applied strong data augmentations, such as Random Augment [21] and Mixup [22] to train ViTs from scratch. They also find AdamW [23] performs better than SGD [24] for training ViTs. Besides, several studies [1, 25, 26] adopted gradient clipping to stabilize and accelerate the convergence of ViTs.

Most related to our work, Steiner et al [27] conducted a comprehensive study on data augmentation and model regularization of ViTs to improve the accuracy. Unlike them, we focus on investigating the strategies to enhance the adversarial robustness.

2.2 Adversarial Robustness

Given a clean example x with its class label y and a model $f_\theta(\cdot)$ with its parameters θ, the goal of an adversary is to find an adversarial perturbation δ that fools the network into making an incorrect prediction for the perturbed image (*i.e.*, $f_\theta(x+\delta) \neq y$), while the perturbation norm does not exceed ϵ (*i.e.*, $\|\delta\|_\infty \leq \epsilon$),

$$\delta = \underset{\|\delta\|_\infty \leq \epsilon}{\arg\max}\, \mathcal{L}(f_\theta(x+\delta), y), \tag{1}$$

while $\mathcal{L}(\cdot,\cdot)$ is usually the cross entropy loss. Previous studies [14, 28, 29, 15, 30] proposed several methods to generate adversarial examples, and Project Gradient Descent (PGD) [15] becomes the most popular one, which is

$$\delta \leftarrow \Pi_\epsilon\big(\delta + \alpha \cdot \text{sign}\big(\nabla_\delta \mathcal{L}(f_\theta(x+\delta), y)\big)\big). \tag{2}$$

To defend against adversarial attacks, while several defense methods [31, 32] are proposed, only adversarial training (AT) [15] and its variants [16, 17, 18, 33] are demonstrated to show notable empirical robustness [34, 35, 36], which directly incorporates adversarial examples into training as follows,

$$\theta = \arg\min_\theta \frac{1}{N} \sum_{i=1}^{N} \max_{\|\delta_i\|_\infty \leq \epsilon} \mathcal{L}(f_\theta(x_i+\delta_i), y_i). \tag{3}$$

Recently, AT of CNNs has become more understood [15, 37, 19]. However, few robustness studies focus on ViT, the brand new architecture in computer vision. For example, several studies [38, 39, 12] have shown ViTs are more robust against common corruptions, which are simpler than adversarial perturbations. Naturally trained ViTs are only able to provide limited robustness against extremely small adversarial perturbations (*e.g.*, under threat model of $\epsilon = 0.001$ on ImageNet[1]) [38, 39], while

[1]The most popular threat model on ImageNet is $\epsilon = 4/255 \approx 0.0157$.

adversarially trained ones fail in providing more adversarial robustness on popular threat models compared to CNNs [12]. Further, AT is extremely sensitive to training settings [19], while there is no training strategy evaluation for AT of ViTs. This motivates us to comprehensively evaluate various training strategies across several datasets, thus bringing the first implementation benchmark for adversarial training of ViTs.

3 Strategies for Adversarial Training of ViTs

In this section, we comprehensively evaluate several training techniques in terms of data (pre-training and data augmentation) and training (optimizer, learning rate schedule, and gradient clipping) across various architectures (vanilla ViT and Swin) and datasets (CIFAR-10 and Imagenette), so as to benchmark the state-of-the-art (SOTA) adversarial robustness of ViTs. All experiments are performed on GPUs of Tesla V100.

Datasets. Since AT is time-consuming (usually $10\times$ slower than natural training), small datasets remain the most popular on adversarial robustness [36]. We adopt CIFAR-10 [40] and Imagenette[2] (a subset of 10 classes from ImageNet-1K) datasets.

Architectures. We use the vanilla ViT (ViT-B) [1] and Swin-B [2] as baseline models. ViT first successfully introduced Transformers from natural language processing [8, 41, 42, 43] to computer vision and achieved competitive recognition performance compared to CNNs. With the help of multi-stage hierarchical architecture, Swin became a general-purpose backbone and achieved SOTA performance on several downstream tasks[3]. By default, all models are pretrained in ImageNet-1K[4]. Considering that the image size on ImageNet is 224×224 and the image size on CIFAR-10 is 32×32, we downsample the weights for patch embeddings and use the patch size 4×4 (the length of the patch sequence becomes $8 \times 8 = 64$ on CIFAR-10) to keep the input size on CIFAR-10.

Perturbation Threat Models. We apply the commonly-used ℓ_∞ attack, that is, the perturbation δ satisfies $\|\delta\|_\infty \leq \epsilon$. Following the most popular threat models [36], we set $\epsilon = 8/255$ for CIFAR-10 and $\epsilon = 8/255$ for Imagenette. When evaluating the adversarial robustness, we apply 20-step PGD (PGD-20) [15] with the step size $2/255$ and Auto-Attack (AA) [35] that is the strongest attack to verify the empirical robustness via an ensemble of diverse attacks (*i.e.*, two variants of PGD attack, FAB attack [44], and Square Attack [45]).

Training Settings. All models are pre-trained on ImageNet-1K and are adversarially trained for 40 epochs using SGD with momentum 0.9, weight decay 1e-4, and an initial learning rate 0.1. The learning rate is divided by 10 at the 36-th and 38-th epoch. Simple data augmentations such as random crop with padding and random horizontal flip are applied. During adversarial training, we use PGD-10 with step size $2/255$ to craft adversarial examples.

3.1 The Necessity of Gradient Clipping

First of all, we find that it is difficult to adversarially train ViTs on small datasets. For example, the final robustness of ViT-B is only 10% on CIFAR-10 and 9.86% on Imagenette in Table 1. To explore the problem, we visualize the norm of gradient during adversarial training in Figure 1, and find the gradient explodes at the very beginning of training. We adopt gradient clipping [46] to AT of ViTs. Specifically, we ensure the maximum of gradient vector is 1 under the l_2-norm. In Table 1, with the help of gradient clipping, ViT-B achieves the robustness of 48.33% and 56.00% on CIFAR-10 and Imagenette. Although Swin-B can be adversarially trained directly, gradient clipping still improves the performance by a notable margin, *i.e.*, improving natural accuracy by 3.52% and robustness by 3.87% on Imagenette. Interestingly, gradient clipping has almost no effect on the results of CNNs (refer to Appendix B), which demonstrates the difference between ViTs and CNNs in terms of training dynamics. Since gradient clipping is almost necessary and beneficial, we always apply it by default in our following experiments.

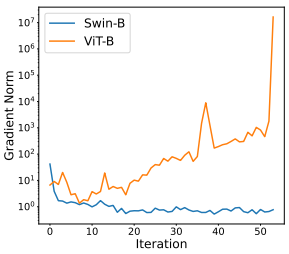


Figure 1: The ℓ_2-norm of gradients during AT of ViTs.

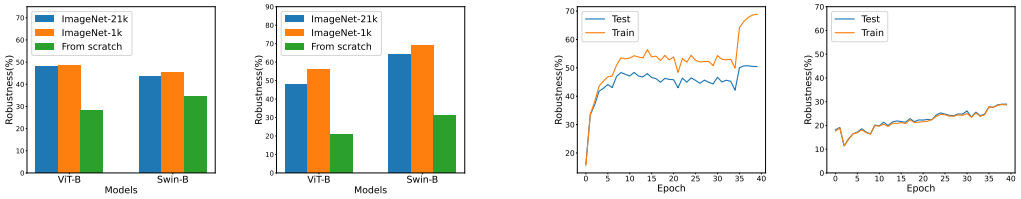
[2] https://github.com/fastai/imagenette

[3] https://paperswithcode.com/sota/object-detection-on-coco

[4] https://github.com/rwightman/pytorch-image-models

Table 1: The performance of adversarially trained ViT-B and Swin-B with or without gradient clipping (GC) on defending against adversarial examples, which are crafted by Auto-Attack.

Model	CIFAR-10		Imagenette	
	Natural	AA	Natural	AA
ViT-B	10.00	10.00	9.86	9.86
ViT-B+GC	**85.53**	**48.33**	**88.08**	**56.00**
Swin-B	82.42	44.90	91.13	65.43
Swin-B+GC	**83.75**	**45.28**	**94.65**	**69.30**



(a) Final robustness of different pre-training strategies (*Left*: CIFAR-10, *Right*: Imagenette)

(b) Learning curve on CIFAR-10 (*Left*: AT of ViT-B pre-trained on ImageNet-1k, *Right*: AT from scratch)

Figure 2: Effects of different pre-training strategies on AT of ViTs.

3.2 The Necessity of Pre-training

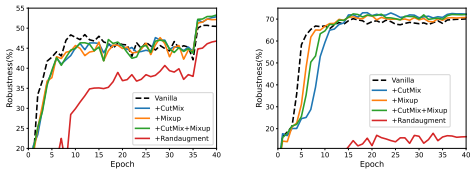
Despite that ViTs perform competitively on ImageNet-1K, which is a difficult recognition task with 1000 classes, their performance heavily relies on the large-scale pre-training [1, 47] and strong data augmentations [3] due to the lack of CNN-like inductive biases. Considering that the input size on CIFAR-10 is much smaller (32×32) and the task itself is much simpler with just 10 classes, we are wondering if pre-training is still necessary for ViTs on such a simple recognition task (although under adversarial attacks).

We evaluate 3 pre-training strategies: 1) pre-training on ImageNet-21K (an extremely large dataset with 14,197,122 images from 21,841 classes), 2) pre-training on ImageNet-1K (a large dataset with more than 1.2M images from 1,000 classes), and 3) no pre-training (*i.e.*, adversarial training from scratch). The final robustness under AA of different AT variants is shown in Figure 2(a). Compared to AT from scratch, pre-training-based AT can improve the robustness of ViTs by a notable margin ($\sim 16\%$ on average). Interestingly, the benefit from pre-training to ViTs is much more significant than those to CNNs [48] (+11.6%), which reflects the stronger necessity of ViTs for pre-training compared to CNNs. Moreover, pre-training on a larger dataset is unable to bring higher final robustness.

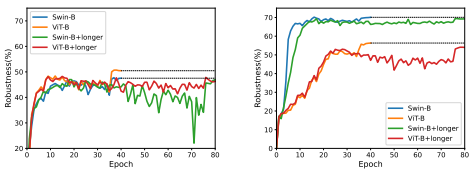
To explore the reason why pre-training can help AT, we plot the learning curve of ViT-B with and without pre-training. In Figure 2(b), we find that ViT without pre-training fails in fitting the training data during adversarial training (the training robustness is only 27.96%). This suggests that the current random initialization for weight parameters of ViTs is not suitable for AT, even on small datasets like CIFAR-10, while pre-training on large datasets could provide a good initialization.

3.3 The Effect of Data Augmentations

In AT of CNNs, a majority of advanced data augmentations like Mixup [22] and RandAugment [21] are proven to be unsuccessful (without considering model weight averaging) [37, 49], which is contrary to natural training. Here, we explore if data augmentations can improve robustness for AT of ViTs. Specifically, we incorporate CutMix [50], Mixup, and Randaugment to AT of ViTs respectively. In Figure 3(a), ViT-B obtains higher robustness on CIFAR-10 with CutMix or Mixup or both, while behaving worse with RandAugment. We also find the similar phenomena of Swin-B on Imagenette. We conjecture this is because RandAugment is a more advanced technique, which is too difficult for AT of ViTs. In conclusion, a suitable combination of data augmentations can directly improve the adversarial robustness of ViTs, which is different from CNNs. In the following experiments, we always adopt CutMix and Mixup to AT of ViTs by default.



(a) Robustness with different data augmentations (*Left*: ViT-B on CIFAR-10, *Right*: Swin-B on Imagenette)



(b) Robustness for training more epochs (*Left*: CIFAR-10, *Right*: Imagenette)

Figure 3: Learning curves of AT with different advanced data augmentations or learning rate schedules. The robustness is evaluated under PGD-20.

Table 2: The performance of adversarially trained ViT-B and Swin-B based on different optimizers.

Dataset	Optimizer	ViT-B		Swin-B	
		Natural	AA	Natural	AA
CIFAR-10	AdamW+cyclic	78.67	46.16	78.33	45.20
	SGD+piecewise	**83.16**	**49.06**	**83.36**	**46.89**
Imagenette	AdamW+cyclic	89.02	61.44	91.16	67.44
	SGD+piecewise	**91.08**	**61.73**	**94.80**	**70.83**

3.4 The Effect of the Optimizer

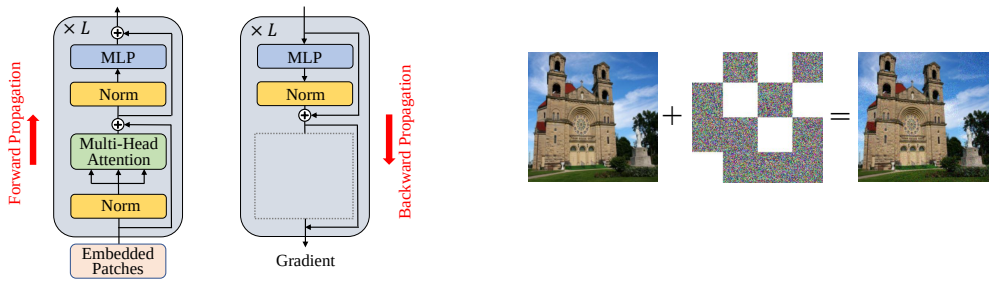
Recalling that SGD achieves the best adversarial robustness in CNNs [15, 37]. By contrast, the default optimizer of ViTs is AdamW [23]. In this part, we study the choice of optimizers on adversarial robustness. We also consider the influence of different combination of optimizer and learning rate scheduler in Appendix C. Part of results on two popular settings are shown in Table 2: 1) AdamW + cyclic learning rate schedule, and 2) SGD + piecewise learning rate schedule. We find SGD + piecewise learning rate schedule helps ViT-B to improve the natural accuracy by 4.51% and the robustness by 2.90% over AdamW + cyclic learning rate schedule on CIFAR-10. In addition, from Appendix C, we can see that AdamW + piecewise learning rate schedule for AT of ViTs obtains worse robustness compared to SGD + piecewise learning rate, which indicates the performance improvements come from the optimizer itself, rather than the learning rate schedule, i.e., SGD consistently improves robustness for ViTs.

3.5 Training for More Epochs

Recent research [51, 52] demonstrated that training for longer can also be viewed as increasing model complexity. Considering that AT requires models with larger complexity [15], we explore if training longer is able to improve adversarial robustness. We adversarially train models for 80 epochs (40 epochs in our default setting) and decay the learning rate at 75-th and 78-th epoch. Figure 3(b) illustrate that robustness has already decreased before the first learning rate decay. Because the robust overfitting of ViTs happens much earlier than those of CNNs, training longer fails in improving the robustness and hurts the final robustness instead.

3.6 Recipe for Adversarial Training of ViTs

After conducting a comprehensive evaluation of different training techniques, we provide a practical recipe for AT of ViTs. First, gradient clipping and pre-training are almost necessary. Second, it is a good choice to use advanced data augmentations (*e.g.*, Cutmix and Mixup), because a suitable combination of data augmentations could improve robustness effectively. Third, in terms of adversarial robustness of ViTs, it is better to use SGD with piecewise learning rate schedule, since it outperforms AdamW with cyclic learning rate schedule (the widely used optimizer for natural training of ViTs). Fourth, since robust overfitting on ViTs happens easier compared to CNNs, it is beneficial to use early stopping.



(a) ARD removes the gradients flowing through the multi-head attention modules with probability p during the back propagation while keeping forward propagation unchanged.

(b) PRM randomly masks a proportion of perturbation with probability during the forward propagation while keeping backward propagation unchanged.

Figure 4: The diagram of our proposed method to improve the adversarial robustness for ViTs.

4 An Architectural Perspective for Adversarially Robust ViTs

In contrast to CNNs, ViTs are a completely different architecture. For example, it divides the images into sequences of image patches before feeding them into the model, and it directly integrates global information across the entire images even in the lowest layers using self-attention [1]. This fundamental difference naturally raises a question whether or not we are able to further improve the robustness based on such specific architectural information. In this section, we explore the impact of these two most obvious architectural changes (multi-head attention and patch-based image splitting) inside ViTs on the adversarial robustness.

4.1 Attention Random Droppping (ARD)

In ViTs, embedded patches[5] are repeatedly processed by encoder blocks, any of which consist of a multi-head attention and a MLP layer, as shown in Figure 4(a). For the i-th block, its input z_{i-1} is first processed by a multi-head attention $\mathcal{A}^{(i)}(\cdot)$, and intermediate result[6] $z'_i = \mathcal{A}^{(i)}(z_{i-1}) + z_{i-1}$ is further processed by the remaining parts (including a MLP layer and a skip connection) of this block $f^{(i)}(\cdot)$. For a ViT with L blocks, its output of the last (*i.e.*, L-th) block is

$$z_L = f^{(L)}(z'_L) = f^{(L)}(\mathcal{A}^{(L)}(z_{L-1}) + z_{L-1}). \tag{4}$$

According to the chain rule in calculus, the gradient of a loss function $\mathcal{L}$ with respect to the input of the last block, z_{L-1}, can then be decomposed as

$$\frac{\partial \mathcal{L}}{\partial z_{L-1}} = \frac{\partial \mathcal{L}}{\partial z_L}\frac{\partial z_L}{\partial z_{L-1}} = \frac{\partial \mathcal{L}}{\partial z_L}\frac{\partial f^{(L)}}{\partial z'_L}(\frac{\partial \mathcal{A}^{(L)}}{\partial z_{L-1}} + 1). \tag{5}$$

Here, we propose the *Attention Random Dropping* (ARD), which randomly drops the gradient flowing the multi-head attention $\mathcal{A}^{(L)}(\cdot)$,

$$\frac{\partial \mathcal{L}}{\partial z_{L-1}} = \frac{\partial \mathcal{L}}{\partial z_L}\frac{\partial f^{(L)}}{\partial z'_L}(u_L \cdot \frac{\partial \mathcal{A}^{(L)}}{\partial z_{L-1}} + 1), \tag{6}$$

where we set u_L to 0 with the probability of p and set it to 1 otherwise. We continue to use the chain rule to obtain the gradient of ARD with respect to the input image x,

$$\frac{\partial \mathcal{L}}{\partial x} = \frac{\partial \mathcal{L}}{\partial z_L}\left(\prod_{i=1}^{L}\frac{\partial f^{(i)}}{\partial z'_i}(u_i \cdot \frac{\partial \mathcal{A}^{(i)}}{\partial z_{i-1}} + 1)\right)\frac{\partial z_0}{\partial x}, \tag{7}$$

where z_0 is the embedded pathes and fed into the 1-st Transformer encoder, and all random variables u_i are *i.i.d.* Figure 4(a) also illustrates the backward propagation using ARD.

[5]To obtain embedded patches from an RGB image, we split the image into fixed-size patches and linearly embed each of them with position embeddings.

[6]Here, the skip connection is explicitly expressed in the formula.

When ARD meets adversarial training, we apply ARD as a warming-up strategy. After the warming-up periods, the training becomes the vanilla AT. In particular, the parameter p in ARD decreases linearly from 1 to 0 during the first n_w epochs, *i.e.*, $p = 1 - n/n_w$ where n is the current epoch ($n \leq n_w$). Here, p is a dynamic parameter along with the training process. At the beginning, relatively weak adversarial samples (large p) are used to warm up while as the training continues, the probability p should be gradually smaller to 0 to ensure the strength of the generated adversarial samples during the whole adversarial training process.

4.2 Perturbation Random Masking (PRM)

ViTs always split the RGB images into non-overlapping patches, and then process them further. Considering this patch-based image splitting, we introduce the *Perturbation Random Masking* (PRM). Specifically, the perturbation on a patch will be masked with probability k during adversarial example generation:

$$\begin{aligned} &\delta' \leftarrow M \cdot \delta, \\ &\delta \leftarrow \Pi_\epsilon\Big(\delta' + \alpha \cdot \text{sign}\big(\frac{\partial \mathcal{L}(f(x+\delta'), y)}{\partial \delta'}\big)\Big). \end{aligned} \tag{8}$$

where δ is the adversarial perturbation, and M is the mask to remove the perturbations on some patches. The second equation is exactly the update rule of PGD attack.

Similar to ARD, we incorporate PRM as a warming-up strategy. Considering an image that is splitted into J patches, the perturbations on $\lfloor J \cdot k \rfloor$ patches will be masked for each iteration during adversarial example generation within AT, and we linearly decrease k from 1 to 0 at the first n_w epochs, i.e., $k = 1 - n/n_w$ where n is the current epoch ($n \leq n_w$). Here, same to p, k is also a dynamic parameter along with the training process.

4.3 Overall Algorithm

Obviously, ARD and PRM are two individual methods that have impact on two different aspects, i.e., ARD influences the backward propagation while PRM influences the update of adversarial perturbations. Thus, we can easily combine them as a mixed warming-up strategy[7]:

$$\begin{aligned} &\delta' \leftarrow M \cdot \delta, \\ &\delta \leftarrow \Pi_\epsilon\Big(\delta' + \alpha \cdot \text{sign}\big(\frac{\partial \mathcal{L}}{\partial z_L}\big(\prod_{i=1}^{L} \frac{\partial f^{(i)}}{\partial z'_i}(u_i \frac{\partial \mathcal{A}^{(i)}}{\partial z_{i-1}} + 1)\big)\frac{\partial z_0}{\partial \delta'}\big)\Big). \end{aligned} \tag{9}$$

The details of ARD or PRM based adversarial training for ViTs are summarized in the Algorithm 1 in Appendix D.

4.4 Evaluation on Benchmark Datasets

In this section, we evaluate the effectiveness of ARD and PRM for the adversarial robustness of ViTs on benchmark datasets.

Experimental Settings. For models, we apply 4 variants of ViT (vanilla ViT [1], DeiT [3], ConViT [53], and Swin [2]) and 3 kinds of scales (base, small, and tiny) for each respectively. For robustness evaluation, we also adopt 20 steps CW [29] (ℓ_∞ version of CW loss optimized by PGD-20) and 100 steps PGD$_\infty$[15] in addition to PGD-20 and AA. We set n_w to 10. We evaluate the performance on: 1) vanilla setting (the basic setting determined in Section 3), 2) the setting with ARD, 3) the setting with PRM, and 4) the setting with ARD and PRM.

Experimental Results. In Table 3, after comparing the same kind of ViTs with different scales, we observe that increasing the size of ViTs can improve both the accuracy on clean examples and the robustness against adversarial attacks. Moreover, diving into each ViT model with different architectural approaches, we find that our proposed ARD or PRM can boost both robustness and accuracy. When only considering the robustness improvement, applying ARD or PRM (one of them) is enough. Taking Swin-S on CIFAR-10 as an example, we can see that the gain for ARD is 1.16%

[7]We denote $\mathcal{L}(f(x+\delta'), \text{y})$ as $\mathcal{L}$ for simplicity.

Table 3: Performance (%) of ARD and PRM with different ViT variants on benchmark datasets. Note that scales like 'B' denotes base, 'S' denotes 'small', and 'Ti' denotes 'tiny'. Here we do not report the results of ViT-Ti and DeiT-B because ViT-Ti does not exist and the architecture of DeiT-B is the same with ViT-B. Top-2 best results are **bold**.

Model	Method	CIFAR-10					Imagenette				
		Natural	CW-20	PGD-20	PGD-100	AA	Natural	CW-20	PGD-20	PGD-100	AA
ViT-S	vanilla	79.59	48.22	50.86	50.73	46.37	78.96	51.90	53.94	53.68	50.60
	+ARD	81.70	**49.07**	**51.72**	51.42	47.12	88.31	**59.16**	**59.21**	**58.80**	**58.24**
	+PRM	**81.77**	49.03	51.67	**51.46**	**47.22**	**88.64**	**59.26**	**59.44**	**58.83**	**58.14**
	+both	**81.96**	**49.04**	**51.68**	**51.43**	**47.19**	**88.94**	58.73	59.06	58.45	57.91
ViT-B	vanilla	83.16	51.11	52.98	52.71	49.06	91.08	63.13	63.63	63.14	61.73
	+ARD	84.21	51.61	53.41	53.10	49.65	92.56	**65.94**	**65.63**	**65.07**	**64.61**
	+PRM	**84.31**	**52.16**	**53.79**	**53.47**	**50.01**	**93.12**	**65.99**	**65.58**	**64.99**	**64.46**
	+both	**84.76**	**51.80**	**53.64**	**53.36**	**49.72**	**93.45**	65.91	65.15	64.51	64.13
DeiT-Ti	vanilla	75.46	45.40	48.10	47.96	43.62	78.96	51.9	53.94	53.68	50.60
	+ARD	77.95	47.28	49.41	49.21	45.45	85.48	**56.97**	**57.15**	**56.87**	**55.77**
	+PRM	**79.09**	**47.64**	**49.76**	**49.53**	**45.68**	**86.50**	56.66	56.69	56.31	55.44
	+both	**79.60**	**48.04**	**50.33**	**50.15**	**45.99**	**86.24**	**56.99**	**56.89**	**56.54**	**55.80**
DeiT-S	vanilla	81.43	49.53	51.88	51.75	47.40	88.69	60.13	60.05	59.69	59.18
	+ARD	81.76	49.49	51.65	51.43	47.55	88.97	**61.53**	**61.25**	**60.82**	**60.03**
	+PRM	**83.02**	**50.47**	**52.50**	**52.26**	**48.27**	**89.45**	60.94	60.76	60.41	59.97
	+both	**83.07**	**50.27**	**52.26**	**52.04**	**48.19**	**90.06**	**61.35**	**61.20**	**60.66**	**60.28**
ConViT-Ti	vanilla	53.09	30.87	33.63	33.61	29.65	51.77	31.85	34.27	34.11	31.34
	+ARD	**79.87**	**47.54**	**50.14**	**49.90**	**45.63**	**86.57**	**58.65**	**58.34**	**57.91**	**57.30**
	+PRM	76.78	45.55	48.14	47.98	43.60	73.99	45.66	48.03	47.82	44.43
	+both	**80.21**	**47.32**	**49.83**	**49.55**	**45.29**	**87.06**	**59.13**	**58.50**	**58.27**	**57.48**
ConViT-S	vanilla	54.03	31.73	34.61	34.60	30.60	80.76	52.71	54.62	54.42	51.69
	+ARD	**84.06**	**50.83**	52.72	52.44	48.71	90.75	**63.95**	**63.72**	**63.26**	**62.68**
	+PRM	84.05	50.79	**52.96**	**52.56**	**48.72**	**90.88**	**64.28**	**63.77**	**63.49**	**62.88**
	+both	**84.38**	**51.00**	**52.85**	**52.47**	**48.86**	**91.57**	63.17	62.65	62.09	61.07
ConViT-B	vanilla	61.54	35.63	38.77	38.71	34.21	90.52	64.2	64.36	64.00	63.44
	+ARD	85.36	**51.51**	**53.16**	**52.96**	49.16	92.13	66.27	66.04	65.58	**65.12**
	+PRM	**85.48**	51.48	52.83	52.48	**49.28**	**92.28**	**66.75**	**66.57**	**65.76**	64.51
	+both	**85.48**	**51.53**	**53.07**	**52.87**	**49.21**	**92.46**	**67.03**	**66.85**	**66.27**	**65.73**
Swin-Ti	vanilla	79.34	45.74	47.95	47.75	43.98	92.64	67.24	66.73	66.47	65.78
	+ARD	78.52	45.70	48.00	47.91	43.84	92.84	67.31	66.93	66.55	65.96
	+PRM	**81.94**	**46.93**	**48.56**	**48.35**	**44.91**	**94.04**	**68.23**	**68.15**	**67.67**	**67.21**
	+both	**82.63**	**47.61**	**48.87**	**48.62**	**45.31**	**94.14**	**68.82**	**68.10**	**67.69**	**67.34**
Swin-S	vanilla	79.34	46.56	48.53	48.32	44.88	93.83	69.35	69.74	68.28	67.85
	+ARD	82.07	47.84	49.56	49.31	**46.04**	94.55	69.73	69.15	68.66	68.23
	+PRM	**84.24**	**48.17**	**49.63**	**49.38**	46.01	**95.16**	**70.22**	**69.68**	**69.10**	**68.51**
	+both	**84.46**	**48.52**	**50.02**	**49.66**	**46.17**	**95.17**	**70.22**	**69.66**	**69.15**	**68.56**
Swin-B	vanilla	83.36	48.22	50.19	49.88	46.89	94.75	70.88	70.06	69.48	69.04
	+ARD	81.24	47.64	49.19	48.83	44.38	95.52	71.39	70.50	69.38	69.30
	+PRM	**84.07**	**49.68**	**50.95**	**50.66**	**47.25**	**96.31**	**74.11**	**73.43**	**72.41**	**71.77**
	+both	**84.16**	**49.78**	**51.47**	**51.19**	**47.50**	**96.18**	**73.30**	**72.05**	**71.41**	**70.62**

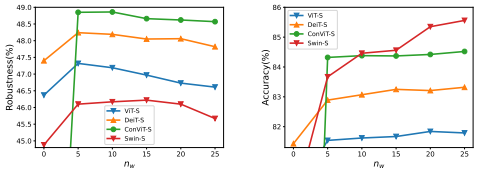
while 1.13% for PRM under AA. When additionally considering accuracy improvement, it is better to use both ARD and PRM. For example, on CIFAR-10 with ViT-B model, applying both ARD and PRM improves accuracy of 1.60% compared with vanilla, while only applying ARD or PRM, the gains are at most 1.15%. Similar results can also be observed on Imagenette.

4.5 Combination with Other Defense Methods

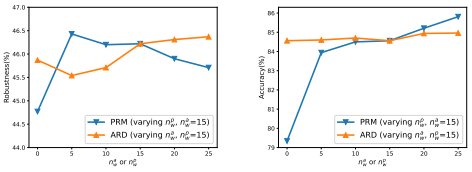
Although the above discussions are mainly concentrated on standard adversarial training, we further demonstrate that our methods: ARD and PRM, can be easily combined with other stronger defense methods. We adopt TRADES [16] and MART [54] here because of their outstanding performance evaluated in [36]. The settings are the same as the Section 4.4 except that we do not apply CutMix and Mixup for data augmentations to keep in line with their original papers. The experiments are conducted on DeiT-Ti, ConViT-Ti and Swin-Ti. As shown in Table 4, the robustness and accuracy of TRADES and MART are improved on both datasets. For example, for DeiT-Ti on CIFAR-10, AA robustness increases by 0.77% on TRADES and 1.63% on MART. It demonstrates that ARD and PRM can further improve the performance of existing defense methods.

Table 4: Performance (%) of our proposed ARD and PRM when combining with other defense methods. The best results are in **bold**.

Model	Method	CIFAR-10					Imagenette				
		Natural	CW-20	PGD-20	PGD-100	AA	Natural	CW-20	PGD-20	PGD-100	AA
DeiT-Ti	TRADES	78.70	46.78	49.63	49.58	46.25	81.63	55.16	56.28	56.08	54.29
	+Ours	**80.24**	**47.60**	**51.02**	**50.97**	**47.02**	**83.75**	**56.94**	**56.76**	**56.56**	**55.87**
	MART	71.7	45.95	49.52	49.37	44.34	75.03	50.04	52.82	52.71	48.89
	+Ours	**74.89**	**47.60**	**51.18**	**51.16**	**45.97**	**83.21**	**56.10**	**56.66**	**56.08**	**53.78**
ConViT-Ti	TRADES	77.70	45.09	48.71	48.63	44.65	73.86	45.83	49.20	49.12	45.40
	+Ours	**80.02**	**47.33**	**50.10**	**50.08**	**46.75**	**85.12**	**58.62**	**58.85**	**58.34**	**57.32**
	+MART	63.68	38.97	42.80	42.77	37.62	50.01	31.87	35.11	35.06	30.93
	+Ours	**74.89**	**47.60**	**51.18**	**51.16**	**45.97**	**83.21**	**56.10**	**56.66**	**56.08**	**53.78**
Swin-Ti	TRADES	79.41	46.45	49.3	49.23	45.74	90.37	67.82	67.64	67.26	66.32
	+Ours	**80.71**	**47.11**	**49.79**	**49.74**	**46.36**	**92.08**	**69.58**	**69.45**	**68.64**	**67.62**
	+MART	75.19	46.10	49.82	49.71	44.54	89.91	66.27	65.66	65.15	63.46
	+Ours	**77.37**	**46.98**	**50.44**	**50.28**	**45.28**	**94.24**	**76.56**	**65.99**	**65.61**	**65.12**



(a) Evaluating the sensitivity of n_w on overall approach. (*Left*: Robustness, *Right*: Accuracy)



(b) Evaluating the sensitivity of n_w on each method individually. (*Left*: Robustness, *Right*: Accuracy)

Figure 5: Performance of ViTs under different hyperparameter configurations. The metric for robustness is AA.

4.6 Hyper-Parameter Analysis

For simplicity of our proposed warming-up strategy, we use the same hyper-parameter n_w as the number of warming-up epochs for both ARD and PRM. We firstly analyze this shared hyper-parameter $n_w \in \{0, 5, 10, 15, 20, 25\}$. The experiments are based on four ViT variants (ViT, DeiT, ConViT, and Swin), and the settings are the same as the Section 4.4. AA is used to evaluate the robustness. In Figure 5(a), we significantly improve the robustness as long as adopting our proposed warming-up strategy into AT, *i.e.*, $n_w > 0$. We obtain the highest robustness at the best n_w (*e.g.*, $n_w = 15$ for Swin-S). Besides, the improvements are not sensitive to the hyper-parameter, which allows us to tune n_w easily.

To further exploit the proposed method, we use two individual hyper-parameters n_w^a and n_w^p for ARD and PRM respectively and we peform our experiments on Swin-S. When fixing one (*e.g.*, n_w^a) as 15 (the best one in the last experiment), we change another hyper-parameter (*e.g.*, n_w^p) as $\{0, 5, 10, 15, 20, 25\}$. In Figure 5 (b), for robustness, these two situations show different trends, *i.e.*, a longer warm-up with ARD (the orange line) increase the final robustness, a longer warm-up with PRM (the blue line) decrease the final robustness. We can find a good balance when $n_w^a = n_w^p = 15$. The former has improved the performance enough, and the latter has not decreased too much. As a result, we set $n_w^a = n_w^p = n_w$ by default.

5 Conclusion

In this paper, we conducted comprehensive experiments to investigate training techniques across datasets to bring a first implementation benchmark for AT of ViTs. We also provided some practical suggestions, including using gradient clipping, pre-training on large datasets, employing stronger data augmentations, and preferring SGD rather than AdamW. Besides, taking the architectural information into consideration, we incorporated two simple but effective methods, ARD and PRM, into AT to improve the adversarial robustness further. Extensive experiments demonstrated the effectiveness of our proposed warming-up strategy. We hope our work not only provides the first implementation benchmark for AT of ViTs, but also reminds researchers of the potential of architectural information contained within brand new models such as ViTs.

References

[1] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[2] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[3] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021.

[4] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, volume 25, 2012.

[6] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[9] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6881–6890, 2021.

[10] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[11] Yifan Jiang, Shiyu Chang, and Zhangyang Wang. Transgan: Two pure transformers can make one strong gan, and that can scale up. *Advances in Neural Information Processing Systems*, 34, 2021.

[12] Yutong Bai, Jieru Mei, Alan L Yuille, and Cihang Xie. Are transformers more robust than cnns? *Advances in Neural Information Processing Systems*, 34, 2021.

[13] 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.

[14] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[15] 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.

[16] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*, 2019.

[17] Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. In *NeurIPS*, 2020.

[18] Sven Gowal, Sylvestre-Alvise Rebuffi, Olivia Wiles, Florian Stimberg, Dan Andrei Calian, and Timothy A Mann. Improving robustness using generated data. In *Advances in Neural Information Processing Systems*, 2021.

[19] Tianyu Pang, Xiao Yang, Yinpeng Dong, Hang Su, and Jun Zhu. Bag of tricks for adversarial training. *arXiv preprint arXiv:2010.00467*, 2020.

[20] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852, 2017.

[21] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical data augmentation with no separate search. *arXiv preprint arXiv:1909.13719*, 2(4):7, 2019.

[22] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[23] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.

[24] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[25] Liangqiong Qu, Yuyin Zhou, Paul Pu Liang, Yingda Xia, Feifei Wang, Li Fei-Fei, Ehsan Adeli, and Daniel Rubin. Rethinking architecture design for tackling data heterogeneity in federated learning. *arXiv preprint arXiv:2106.06047*, 2021.

[26] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9640–9649, 2021.

[27] Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision transformers. *arXiv preprint arXiv:2106.10270*, 2021.

[28] Alexey Kurakin, Ian Goodfellow, Samy Bengio, et al. Adversarial examples in the physical world, 2016.

[29] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017.

[30] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193, 2018.

[31] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE symposium on security and privacy (SP)*, pages 582–597. IEEE, 2016.

[32] Xingjun Ma, Bo Li, Yisen Wang, Sarah M Erfani, Sudanthi Wijewickrema, Grant Schoenebeck, Dawn Song, Michael E Houle, and James Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality. *arXiv preprint arXiv:1801.02613*, 2018.

[33] Sylvestre-Alvise Rebuffi, Sven Gowal, Dan A Calian, Florian Stimberg, Olivia Wiles, and Timothy Mann. Fixing data augmentation to improve adversarial robustness. *arXiv preprint arXiv:2103.01946*, 2021.

[34] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*, pages 274–283. PMLR, 2018.

[35] 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.

[36] 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*, 2021.

[37] Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. In *International Conference on Machine Learning*, pages 8093–8104. PMLR, 2020.

[38] Rulin Shao, Zhouxing Shi, Jinfeng Yi, Pin-Yu Chen, and Cho-Jui Hsieh. On the adversarial robustness of visual transformers. *arXiv e-prints*, pages arXiv–2103, 2021.

[39] Shiyu Tang, Ruihao Gong, Yan Wang, Aishan Liu, Jiakai Wang, Xinyun Chen, Fengwei Yu, Xianglong Liu, Dawn Song, Alan Yuille, et al. Robustart: Benchmarking robustness on architecture design and training techniques. *arXiv preprint arXiv:2109.05211*, 2021.

[40] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Tech Report*, 2009.

[41] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[42] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.

[43] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[44] 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.

[45] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *European Conference on Computer Vision*, pages 484–501. Springer, 2020.

[46] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. Understanding the exploding gradient problem. *CoRR, abs/1211.5063*, 2(417):1, 2012.

[47] Samira Abnar, Mostafa Dehghani, Behnam Neyshabur, and Hanie Sedghi. Exploring the limits of large scale pre-training. *arXiv preprint arXiv:2110.02095*, 2021.

[48] Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. In *International Conference on Machine Learning*, pages 2712–2721. PMLR, 2019.

[49] Sven Gowal, Chongli Qin, Jonathan Uesato, Timothy Mann, and Pushmeet Kohli. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *arXiv preprint arXiv:2010.03593*, 2020.

[50] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.

[51] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.

[52] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124003, 2021.

[53] Stéphane d'Ascoli, Hugo Touvron, Matthew L Leavitt, Ari S Morcos, Giulio Biroli, and Levent Sagun. Convit: Improving vision transformers with soft convolutional inductive biases. In *International Conference on Machine Learning*, pages 2286–2296. PMLR, 2021.

[54] Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations*, 2019.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Appendix A.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Appendix A.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] We will release upon acceptance.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 3.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 3.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Section 3.
   (b) Did you mention the license of the assets? [Yes] See Section 3.
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]