SMOOTH ADVERSARIAL TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

It is commonly believed that networks cannot be both accurate and robust, that gaining robustness means losing accuracy. It is also generally believed that, unless making networks larger, network architectural elements would otherwise matter little in improving adversarial robustness. Here we present evidence to challenge these common beliefs by a careful study about adversarial training. Our key observation is that the widely-used ReLU activation function significantly weakens adversarial training due to its non-smooth nature. Hence we propose *smooth adversarial training (SAT)*, in which we replace ReLU with its smooth approximations to strengthen adversarial training. The purpose of smooth activation functions in SAT is to allow it to find harder adversarial examples and compute better gradient updates during adversarial training. Compared to standard adversarial training, SAT improves adversarial robustness for "free", *i.e.*, no drop in accuracy and no increase in computational cost. For example, without introducing additional computations, SAT significantly enhances ResNet-50's robustness from 33.0% to 42.3%, while also improving accuracy by 0.9% on ImageNet. SAT also works well with larger networks: it helps EfficientNet-L1 to achieve 82.2% accuracy and 58.6% robustness on ImageNet, outperforming the previous state-of-the-art defense by 9.5% for accuracy and 11.6% for robustness.

1 INTRODUCTION

Convolutional neural networks can be easily attacked by adversarial examples, which are computed by adding small perturbations to clean inputs (Szegedy et al., 2014). Many efforts have been devoted to improving network resilience against adversarial attacks (Papernot et al., 2016; Guo et al., 2018; Xie et al., 2018; Liu et al., 2018; Pang et al., 2019; Schott et al., 2019). Among them, adversarial training (Goodfellow et al., 2015; Kurakin et al., 2017; Madry et al., 2018), which trains networks with adversarial examples on-the-fly, stands as one of the most effective methods. Later works further improve adversarial training by feeding networks with harder adversarial examples (Wang et al., 2019), maximizing the margin of networks (Ding et al., 2020), optimizing a regularized surrogate loss (Zhang et al., 2019), *etc*. While these methods achieve stronger adversarial robustness, they sacrifice accuracy on clean inputs. It is generally believed this trade-off between accuracy and robustness might be inevitable (Tsipras et al., 2019), unless additional computational budgets are introduced to enlarge network capacities, *e.g.*, making wider or deeper networks (Madry et al., 2018; Xie & Yuille, 2020), adding denoising blocks (Xie et al., 2019).

Another popular direction for increasing robustness against adversarial attacks is gradient masking (Papernot et al., 2017; Athalye et al., 2018), which usually introduces non-differentiable operations (e.g., discretization (Buckman et al., 2018; Rozsa & Boult, 2019)) to obfuscate gradients. With degenerated gradients, attackers cannot successfully optimize the targeted loss and fail to break such defenses. Nonetheless, gradient masking will be ineffective if its differentiable approximation is used for generating adversarial examples (Athalye et al., 2018).

The bitter history of gradient masking defenses motivates us to rethink the relationship between gradient quality and adversarial robustness, especially in the context of adversarial training where gradients are applied more frequently than standard training. In addition to computing gradients to update network parameters, adversarial training also requires gradient computation for generating training samples. Guided by this principle, we identify that ReLU, a widely-used activation function in most network architectures, significantly weakens adversarial training due to its non-smooth nature, *e.g.*, ReLU's gradient gets an abrupt change when its input is zero, as illustrated in Figure 1.

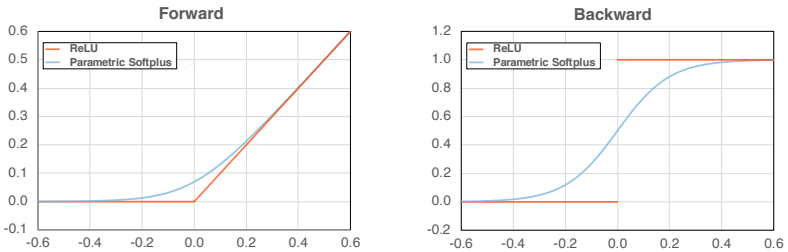

Figure 1: *Left panel*: ReLU and Parametric Softplus. *Right panel*: the first derivatives for ReLU and Parametric Softplus. Compared to ReLU, Parametric Softplus is smooth with continuous derivatives.

To fix the issue induced by ReLU, we propose smooth adversarial training (SAT), which enforces architectural smoothness via replacing ReLU with its smooth approximations[1] for improving the gradient quality in adversarial training (Figure 1 shows Parametric Softplus, an example of smooth approximations for ReLU). With smooth activation functions, SAT is able to feed the networks with harder adversarial training samples and compute better gradient updates for network optimization, hence substantially strengthens adversarial training. Our experiment results show that SAT improves adversarial robustness for "free", *i.e.*, without incurring additional computations or degrading standard accuracy. For instance, by training with the economical *single-step PGD attacker* (*i.e.*, FGSM attacker with random initialization)[2] on ImageNet (Russakovsky et al., 2015), SAT significantly improves ResNet-50's robustness by 9.3%, from 33.0% to 42.3%, while increasing the standard accuracy by 0.9% without incurring additional computational cost.

We also explore the limits of SAT with larger networks. We obtain the best result by using EfficientNet-L1, which achieves 82.2% accuracy and 58.6% robustness on ImageNet, significantly outperforming the prior art (Qin et al., 2019) by 9.5% for accuracy and 11.6% for robustness.

2 Related Works

Adversarial training. Adversarial training improves robustness by training models on adversarial examples (Szegedy et al., 2014; Goodfellow et al., 2015; Kurakin et al., 2017; Madry et al., 2018). Existing works suggest that, to further adversarial robustness, we need to either sacrifice accuracy on clean inputs (Wang et al., 2019; 2020; Zhang et al., 2019; Ding et al., 2020), or incur additional computational cost (Madry et al., 2018; Xie & Yuille, 2020; Xie et al., 2019). This phenomenon is referred to as *no free lunch in adversarial robustness* (Tsipras et al., 2019; Nakkiran, 2019; Su et al., 2018). In this paper, we show that, with SAT, adversarial robustness can be improved for "free".

Our work is also related to the theoretical study (Sinha et al., 2018), which shows replacing ReLU with smooth alternatives can help networks get a tractable bound when certifying distributional robustness. In this paper, we empirically corroborate the benefits of utilizing smooth activations is also observable in the practical adversarial training on the real-world dataset using large networks.

Gradient masking. Besides training models on adversarial data, other ways for improving adversarial robustness include defensive distillation (Papernot et al., 2016), randomized transformations (Xie et al., 2018; Dhillon et al., 2018; Liu et al., 2018; Wang et al., 2018; Bhagoji et al., 2018; Xiao & Zheng, 2020), adversarial input purification (Guo et al., 2018; Prakash et al., 2018; Meng & Chen, 2017; Song et al., 2018; Samangouei et al., 2018; Liao et al., 2018; Bhagoji et al., 2018; Pang et al., 2020), *etc*. Nonetheless, these defense methods degenerate the gradient quality, therefore induce the gradient masking issue (Papernot et al., 2017), which gives a false sense of adversarial robustness (Athalye et al., 2018). In contrast to these works, we aim to improve adversarial robustness by providing networks with better gradients, but in the context of adversarial training.

3 ReLU Weakens Adversarial Training

We hereby perform a series of control experiments in the backward pass of gradient computations to investigate how ReLU weakens, and how its smooth approximation strengthens adversarial training.

[1]More precisely, when we say a function is smooth in this paper, we mean this function is $\mathcal{C}^1$ smooth, *i.e.*, its first derivative is continuous everywhere.

[2]Models trained with single-step PGD attackers only cost $\sim 1.5\times$ training time than standard training

3.1 ADVERSARIAL TRAINING

Adversarial training (Szegedy et al., 2014; Goodfellow et al., 2015; Madry et al., 2018), which trains networks with adversarial examples on-the-fly, aims to optimize the following framework:

$$\arg\min_{\theta} \mathbb{E}_{(x,y)\sim\mathbb{D}}\Big[\max_{\epsilon\in\mathbb{S}} L(\theta, x+\epsilon, y)\Big], \tag{1}$$

where $\mathbb{D}$ is the underlying data distribution, $L(\cdot,\cdot,\cdot)$ is the loss function, θ is the network parameter, x is a training sample with the ground-truth label y, ϵ is the added adversarial perturbation, and $\mathbb{S}$ is the allowed perturbation range. As shown in Equation (1), adversarial training consists of two computation steps: an **inner maximization step**, which computes adversarial examples, and an **outer minimization step**, which computes parameter updates.

Adversarial training setup. We choose ResNet-50 (He et al., 2016) as the backbone network. We apply the PGD attacker (Madry et al., 2018) to generate adversarial perturbations ϵ. Specifically, we select the cheapest version of PGD, *single-step PGD* (PGD-1), to lower the training cost. Following (Shafahi et al., 2019; Wong et al., 2020), we set the maximum per-pixel change $\epsilon = 4$ and the attack step size $\beta = 4$. We follow the basic ResNet training recipes on ImageNet: models are trained for a total of 100 epochs using momentum SGD optimizer, with the learning rate decreased by $10\times$ at the 30-th, 60-th and 90-th epoch; no regularization except a weight decay of 1e-4 is applied.

When evaluating adversarial robustness, we measure the model's top-1 accuracy against the 200-step PGD attacker (PGD-200) on the ImageNet validation set, with the maximum perturbation size $\epsilon = 4$ and the step size $\beta = 1$. We note 200 attack iteration is enough to let PGD attacker converge. Meanwhile, we report the model's top-1 accuracy on the original ImageNet validation set.

3.2 HOW GRADIENT QUALITY AFFECTS ADVERSARIAL TRAINING?

As shown in Figure 1, the widely used activation function, ReLU (Hahnloser et al., 2000; Nair & Hinton, 2010), is non-smooth. ReLU's gradient takes an abrupt change, when its input is 0, therefore significantly degrades the gradient quality. We conjecture that this non-smooth nature hurts the training process, especially when we train models adversarially. This is because, compared to standard training which only computes gradients for updating network parameter θ, adversarial training requires additional computations for the inner maximization step to craft the perturbation ϵ.

To fix this problem, we first introduce a smooth approximation of ReLU, named *Parametric Softplus* (Nair & Hinton, 2010), as $f(\alpha, x) = \frac{1}{\alpha}\log(1+\exp(\alpha x))$, where the hyperparameter α is used to control the curve shape. The derivative of this function w.r.t. the input x is:

$$\frac{d}{dx} f(\alpha, x) = \frac{1}{1+\exp(-\alpha x)} \tag{2}$$

To better approximate the curve of ReLU, we empirically set $\alpha = 10$. As shown in Figure 1, compared to ReLU, Parametric Softplus (α=10) is smooth because it has a continuous derivative.

With Parametric Softplus, we next diagnose how gradient quality in *the inner maximization step* and *the outer minimization step* affects the accuracy and robustness of ResNet-50 in adversarial training. **To clearly benchmark the effects, we only substitute ReLU with Equation (2) in the backward pass, while leaving the forward pass unchanged**, *i.e.*, ReLU is always used for model inference.

Improving gradient quality for the adversarial attacker. We first take a look at the effects of gradient quality on computing adversarial examples (*i.e.*, *the inner maximization step*) during training. More precisely, in the inner step of adversarial training, we use ReLU in the forward pass, but Parametric Softplus in the backward pass; and in the outer step, we use ReLU in both the forward and the backward pass. As shown in the second row of Table 1, when the attacker uses Parametric Softplus's gradient to craft training samples, the resulted model exhibits a performance trade-off compared to the ReLU baseline, *e.g.*, it improves adversarial robustness by 1.5% but degrades accuracy by 0.5%. We note this performance trade-off is also observed when training networks with harder adversarial examples (Wang et al., 2019), therefore motivate us to hypothesize better gradients for the inner maximization step actually boosts the attacker's strength during training. To verify this hypothesis, we evaluate the robustness of two ResNet-50 models via PGD-1 (vs. PGD-200 in Table 1), one with standard training and one with adversarial training. Specifically, during the evaluation, the attacker

	Improving Gradient Quality for the Adversarial Attacker	Improving Gradient Quality for the Network Parameter Updates	Accuracy (%)	Robustness (%)
ResNet-50	✗	✗	68.8	33.0
	✓	✗	68.3 (-0.5)	34.5 (+1.5)
	✗	✓	69.4 (+0.6)	35.8 (+2.8)
	✓	✓	68.9 (+0.1)	36.9 (+3.9)

Table 1: ReLU significantly weakens adversarial training. By improving gradient quality for either the adversarial attacker or the network optimizer, resulted models obtains better robustness than the ReLU baseline. The best robustness is achieved by adopting better gradients for both the attacker and the network optimizer.

uses ReLU in the forward pass, but Parametric Softplus in the backward pass. With better gradients, the PGD-1 attacker is strengthened and hurts models more: it can further decrease the top-1 accuracy by 4.0% (from 16.9% to 12.9%) on the model with standard training and by 0.7% (from 48.7% to 48.0%) on the model with adversarial training (both not shown in Table 1).

Improving gradient quality for network parameter updates. We then study the role of gradient quality on updating network parameters (*i.e.*, *the outer minimization step*) during training. More precisely, in the inner step of adversarial training, we always use ReLU; but in the outer step, we use ReLU in the forward pass, and Parametric Softplus in the backward pass. Surprisingly, this strategy improves adversarial robustness for "free". As shown in the third row of Table 1, without incurring additional computations, adversarial robustness is boosted by 2.8%, and meanwhile accuracy is improved by 0.6%, compared to the ReLU baseline. We note the corresponding training loss also gets lower: the cross-entropy loss on the training set is reduced from 2.71 to 2.59. These results of better robustness and accuracy, and lower training loss together suggest that, with Parametric Softplus in the backward pass of the outer minimization step, networks are able to compute better gradient updates in adversarial training. Interestingly, we also observe that better gradient updates improve the standard training, *i.e.*, with ResNet-50, training with better gradients is able to improve accuracy from 76.8% to 77.0%, and reduces the corresponding training loss from 1.22 to 1.18. These results on both adversarial training and standard training suggest that updating network parameters using better gradients could serve as a principle for improving performance in general, while keeping the inference process of the model unchanged (*i.e.*, ReLU is always used for inference).

Improving gradient quality for both the adversarial attacker and network parameter updates. Given the observation that improving ReLU's gradient for either the adversarial attacker or the network optimizer benefits robustness, we further enhance adversarial training by replacing ReLU with Parametric Softplus in all backward passes, but keeping ReLU in all forward passes. As expected, such a trained model reports the best robustness so far, *i.e.*, as shown in the last row of Table 1, it substantially outperforms the ReLU baseline by 3.9% for robustness. Interestingly, this improvement still comes for "free", *i.e.*, it reports 0.1% higher accuracy than the ReLU baseline. We conjecture this is mainly due to the positive effect on accuracy brought by computing better gradient updates (increase accuracy) slightly overriding the negative effects on accuracy brought by creating harder training samples (hurt accuracy) in this experiment.

3.3 Can Other Training Enhancements Remedy ReLU's Gradient Issue?

More attack iterations. It is known that increasing the number of attack iterations can create harder adversarial examples (Madry et al., 2018). We confirm in our own experiments that by training with the PGD attacker with more iterations, the resulted model exhibits a similar behavior to the case where we apply better gradients for the attacker. By increasing the attacker's cost by $2\times$, PGD-2 improves the ReLU baseline by 0.6% for robustness while losing 0.1% for accuracy. This result suggests we can remedy ReLU's gradient issue *in the inner step of adversarial training* if more computations are given.

Training longer. It is also known longer training lowers the training loss (Hoffer et al., 2017), which we explore next. Interestingly, by extending the default setup to a $2\times$ training cost (*i.e.*, 200 epochs), though the final model indeed achieves a lower training loss (from 2.71 to 2.62), there still exhibits a trade-off between accuracy and robustness. Longer training gains 2.6% for accuracy but loses 1.8% for robustness. On the contrary, our previous experiment shows applying better gradients to optimize networks improves both robustness and accuracy. This discouraging result suggests training longer *cannot* fix the issues *in the outer step of adversarial training* caused by ReLU's poor gradient.

Conclusion. Given these results, we conclude that ReLU significantly weakens adversarial training. Moreover, it seems that the degenerated performance cannot be simply remedied even with training

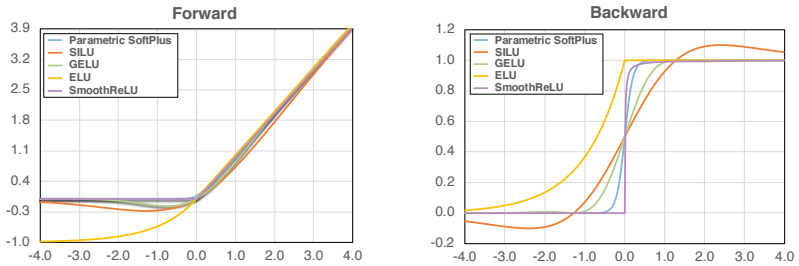

Figure 2: Visualizations of smooth activation functions and their derivatives.

enhancements (*i.e.*, increasing the number of attack iterations & training longer). We identify that the key is ReLU's poor gradient—by replacing ReLU with its smooth approximation *only in the backward pass* substantially improves robustness, even without sacrificing accuracy and incurring additional computational cost. In the next section, we show that making activation functions smooth is a good design principle for enhancing adversarial training in general.

4 Smooth Adversarial Training

As shown above, improving ReLU's gradient can both strengthen the attacker and provide better gradient updates. Nonetheless, this strategy may be suboptimal as there still is a discrepancy between the forward pass (which we use ReLU) and the backward pass (which we use Parametric Softplus). To fully exploit the potential of training with better gradients, we hereby propose smooth adversarial training (SAT), which enforces architectural smoothness via the exclusive usage of smooth activation functions in adversarial training. We keep all other network components the same, as most of them will not result in the issue of poor gradient.[3]

4.1 Adversarial Training with Smooth Activation Functions

We consider the following activation functions as the smooth approximations of ReLU in SAT (Figure 2 plots these functions as well as their derivatives):

- **Softplus** (Nair & Hinton, 2010): $\text{Softplus}(x) = \log(1+\exp(x))$. We also consider its parametric version, *i.e.*, $\frac{1}{\alpha}\log(1+\exp(\alpha x))$, and set $\alpha = 10$ as in Section 3.
- **SILU** (Ramachandran et al., 2017; Elfwing et al., 2018; Hendrycks & Gimpel, 2016): $\text{SILU}(x) = x \cdot \text{sigmoid}(x)$. Compared to others, SILU has a non-monotonic "bump" when $x < 0$.
- **Gaussian Error Linear Unit** (GELU) (Hendrycks & Gimpel, 2016): $\text{GELU}(x) = x \cdot \Phi(x)$, where $\Phi(x)$ is the cumulative distribution function of the standard normal distribution.
- **Exponential Linear Unit** (ELU) (Clevert et al., 2016): if $x \geq 0$, $\text{ELU}(x, \alpha) = x$; otherwise $\text{ELU}(x, \alpha) = \alpha(\exp(x) - 1)$, where we set $\alpha = 1$ as default. Note that when $\alpha \neq 1$, the gradient of ELU is not continuously differentiable anymore. We will be discussing the effects of these non-smooth variants of ELU ($\alpha \neq 1$) on adversarial training in Section 4.3.

Main results. We follow the settings in Section 3 to adversarially train ResNet-50 equipped with smooth activation functions. The results are shown in Figure 3. Compared to the ReLU baseline, all smooth activation functions substantially boost robustness, while keeping the standard accuracy almost the same. For example, smooth activation functions at least boost robustness by 5.7% (using Parametric Softplus, from 33% to 38.7%). We believe such improvement is generalizable to other smooth alternatives (Misra, 2019; Lokhande et al., 2020). Our strongest robustness is achieved by SILU, which enables ResNet-50 to achieve 42.3% robustness and 69.7% standard accuracy.

Additionally, we compare to the setting in Section 3 where Parametric Softplus is only applied at the backward pass. Interestingly, by additionally replacing ReLU with Parametric Softplus at the forward pass, the resulted model further improves robustness by 1.8% (from 36.9% to 38.7%) while keeping the accuracy almost the same, demonstrating the importance of applying smooth activation functions in both forward and backward passes in SAT.

[3]We ignore the gradient issue caused by max pooling, which is also non-smooth, in SAT. This is because modern architectures rarely adopt it, *e.g.* only one max pooling layer is adopted in ResNet (He et al., 2016), and none is adopted in EfficientNet (Tan & Le, 2019).

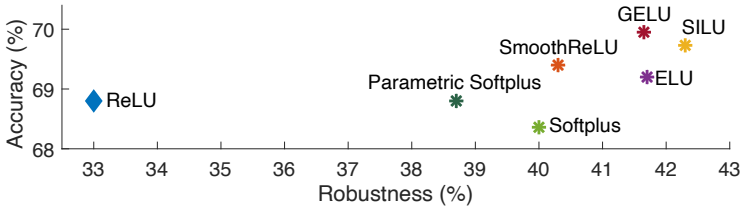

Figure 3: Smooth activation functions improve adversarial training. Compared to ReLU, all smooth activation functions significantly boost robustness, while keeping accuracy almost the same.

α	Robustness (%)	
	ELU	CELU
1	41.1	
1.2	-0.3	+0.1
1.4	-2.0	-0.3
1.6	-3.7	-0.3
1.8	-6.2	-0.2
2.0	-7.9	-0.5

Table 2: Robustness comparison between ELU (non-smooth when $\alpha \neq 1$) and CELU (always smooth $\forall\alpha$).

4.2 Ruling Out the Effect From $x < 0$

Compared to ReLU, in addition to being smooth, the functions above have non-zero responses to negative inputs ($x < 0$) which may also affect adversarial training. To rule out this factor, we hereby propose SmoothReLU, which flattens the activation function by only modifying ReLU after $x \geq 0$,

$$\text{SmoothReLU}(x, \alpha) = \begin{cases} x - \frac{1}{\alpha}\log(\alpha x + 1) & \text{if } x \geq 0, \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

where α is a learnable variable shared by all channels, and is constrained to be positive. We note SmoothReLU is always continuously differentiable regardless the value of α, as

$$\frac{d}{dx}\text{SmoothReLU}(x, \alpha) = \begin{cases} \frac{\alpha x}{1+\alpha x} & \text{if } x \geq 0, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

SmoothReLU converges to ReLU when $\alpha \to \infty$. Note that α needs to be initialized at a large enough value (*e.g.*, 400 in our experiments) to avoid the gradient vanishing problem at the beginning of training. We plot SmoothReLU and its first derivative in Figure 2.

We observe SmoothReLU substantially outperforms ReLU by 7.3% for robustness (from 33.0% to 40.3%), and by 0.6% for accuracy (from 68.8% to 69.4%), therefore clearly demonstrates the importance of a function to be smooth, and rules out the effect from having responses when $x < 0$.

4.3 Case Study: Stabilizing Adversarial Training with ELU using CELU

In the analysis above, we show that adversarial training can be greatly improved by replacing ReLU with its smooth approximations. To further demonstrate the generalization of SAT (beyond ReLU), we discuss another type of activation function—ELU. The first derivative of ELU is shown below:

$$\frac{d}{dx}\text{ELU}(x, \alpha) = \begin{cases} 1 & \text{if } x \geq 0, \\ \alpha\exp(x) & \text{otherwise.} \end{cases} \tag{5}$$

Here we mainly discuss the scenario when ELU is non-smooth, *i.e.*, $\alpha \neq 1$. As can be seen from Equation (5), ELU's gradient is not continuously differentiable anymore, *i.e.*, $\alpha\exp(x) \neq 1$ when $x = 0$, therefore resulting in an abrupt gradient change like ReLU. Specifically, we consider the range $1.0 < \alpha \leq 2.0$, where the gradient abruption becomes more drastic with a larger value of α.

We show the adversarial training results in Table 2. Interestingly, we observe that the adversarial robustness is highly dependent on the value of α—the strongest robustness is achieved when the function is smooth (*i.e.*, $\alpha = 1.0$, 41.4% robustness), and all other choices of α monotonically decrease the robustness when α gradually approaches 2.0. For instance, with $\alpha = 2.0$, the robustness drops to only 33.2%, which is 7.9% lower than that of using $\alpha = 1.0$. The observed phenomenon here is consistent with our previous conclusion on ReLU—non-smooth activation functions significantly weaken adversarial training.

To stabilize the adversarial training with ELU, we apply its smooth version, CELU (Barron, 2017), which re-parametrize ELU to the following format:

$$\text{CELU}(x, \alpha) = \begin{cases} x & \text{if } x \geq 0, \\ \alpha\left(\exp\left(\frac{x}{\alpha}\right) - 1\right) & \text{otherwise.} \end{cases} \tag{6}$$

The first derivatives of CELU can be written as follows:

$$\frac{d}{dx}\,\mathrm{CELU}(x,\alpha) = \begin{cases} 1 & \text{if } x \geq 0, \\ \exp\frac{x}{\alpha} & \text{otherwise.} \end{cases} \tag{7}$$

With this parameterization, CELU is now continuously differentiable regardless of the choice of α.

We observe that CELU greatly stabilizes adversarial training, *i.e.*, compared to $\alpha = 1.0$, the worst case in CELU is merely 0.5% lower (shown in Table 2). Recall that this gap for ELU is 7.9%. This case study provides another strong support on justifying the importance of performing SAT.

5 Exploring the Limits of Smooth Adversarial Training

Recent works (Xie & Yuille, 2020; Gao et al., 2019) show that, compared to standard training, adversarial training exhibits a much stronger requirement for larger networks to obtain better performance. Nonetheless, previous explorations in this direction only consider either deeper networks (Xie & Yuille, 2020) or wider networks (Madry et al., 2018), which might be insufficient. To this end, we hereby present a systematic study on showing how network scaling up behaves in SAT. Specifically, we set SILU as the default activation function to perform SAT, as it achieves the best robustness among different candidates (as shown in Figure 3).

5.1 Scaling-up ResNet

We first perform the network scaling-up experiments with ResNet in SAT. In standard training, (Tan & Le, 2019) suggest that, all three scaling-up factors, *i.e.*, *depth, width and image resolutions*, are important to further improve ResNet performance. We hereby examine the effects of these factors in SAT. We choose ResNet-50 (with the default image resolution at 224) as the baseline network.

Depth & width. Previous works already show that making networks deeper or wider can further standard adversarial training. We re-verify this conclusion in SAT. As shown in the second to fifth rows of Table 3, we confirm that both deeper or wider networks consistently outperform the baseline network in SAT. For instance, by training a deeper ResNet-152, it improves ResNet-50's performance by 4.2% for accuracy and 3.7% for robustness. Similarly, by training a $4\times$ wider ResNeXt-50-32x8d (Xie et al., 2017), it improves accuracy by 3.9% and robustness by 2.8%.

Image resolution. Though larger image resolution benefits standard training, it is generally believed that scaling up this factor will induce weaker adversarial robustness (Galloway et al., 2019). However, surprisingly, this belief is invalid when taking adversarial training into consideration. As shown in the sixth and seventh rows of Table 3, ResNet-50 consistently achieves better performance when training with larger image resolutions in SAT. We conjecture this improvement is possibly due to a larger image resolution (1) enables attackers to create stronger adversarial examples (Galloway et al., 2019); and (2) increases network capacity (Tan & Le, 2019), therefore benefits SAT overall.

Compound scaling. So far, we have confirmed that the basic scaling of depth, width and image resolution are all important scaling-up factors in SAT. As argued in (Tan & Le, 2019) for standard training, scaling up all these factors simultaneously is better than just focusing on a single dimension. To this end, we make an attempt to create a simple compound scaling for ResNet. As shown in the last row of Table 3, the resulted model, ResNeXt-152-32x8d with input resolution at 380, achieves a much stronger result than the ResNet-50 baseline, *i.e.*, +8.5% for accuracy and +8.9% for robustness.

	Accuracy (%)	Robustness (%)
ResNet-50	69.7	42.3
+ 2x deeper (ResNet-101)	72.9 (+3.2)	45.5 (+3.2)
+ 3x deeper (ResNet-152)	73.9 (+4.2)	46.0 (+3.7)
+ 2x wider (ResNeXt-50-32x4d)	71.2 (+1.5)	42.5 (+0.2)
+ 4x wider (ResNeXt-50-32x8d)	73.6 (+3.9)	45.1 (+2.8)
+ larger resolution 299	70.9 (+1.2)	43.8 (+1.5)
+ larger resolution 380	71.6 (+1.9)	44.1 (+1.8)
+ 3x deeper & 4x wider (ResNeXt-152-32x8d) & larger resolution 380	**78.2 (+8.5)**	**51.2 (+8.9)**

Table 3: Scaling-up ResNet in SAT. We observe SAT consistently helps larger networks get better performance.

Discussion on standard adversarial training. We first verify basic scaling of depth, width and image resolution also matter in *standard adversarial training*, *e.g.*, by scaling up ResNet-50 (33.0% robustness), the deeper ResNet-152 achieves 39.4% robustness (+6.4%), the wider ResNeXt-50-32x8d achieves 36.7% robustness (+3.7%), and the ResNet-50 with larger image resolution at 380

achieves 36.9% robustness (+3.9%). Nonetheless, all these robustness performances are lower than the robustness achieved by the SAT's ResNet-50 (42.3%, first row of Table 3). In other words, scaling up networks seems less effective than replacing ReLU with smooth activation functions.

We also find compound scaling is more effective than basic scaling for standard adversarial training, *e.g.*, ResNeXt-152-32x8d with input resolution at 380 here reports 46.3% robustness. Although this result is better than adversarial training with basic scaling above, it is still ~5% lower than SAT with compound scaling, *i.e.*, 46.3% v.s. 51.2%. In other words, even with larger networks, applying smooth activation functions in adversarial training is still essential for improving performance.

5.2 SAT WITH EFFICIENTNET

The results on ResNet show that scaling up networks in SAT effectively improves performance. Nonetheless, the applied scaling policies could be suboptimal, as they are hand-designed without any optimizations. EfficientNet (Tan & Le, 2019), which uses neural architecture search (Zoph & Le, 2016) to automatically discover the optimal factors for network scaling, provides a strong family of models for image recognition. To examine the benefits of EfficientNet, we now use it to replace ResNet in SAT. Note that all other training settings are the same as described in our ResNet experiments.

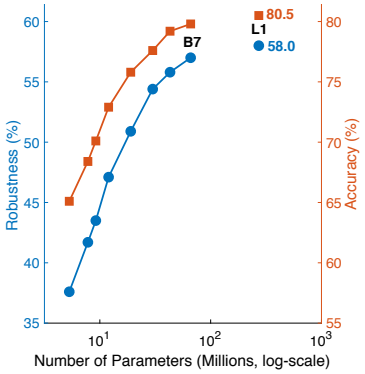

Figure 4: Scaling-up EfficientNet in SAT. Note EfficientNet-L1 is not connected to the rest of the graph because it was not part of the compound scaling suggested by (Tan & Le, 2019).

Similar to ResNet, Figure 4 shows stronger backbones consistently achieve better performance in SAT. For instance, by scaling the network from EfficientNet-B0 to EfficientNet-B7, the robustness is improved from 37.6% to 57.0%, and the accuracy is improved from 65.1% to 79.8%. Surprisingly, the improvement is still observable for larger networks: EfficientNet-L1 (Xie et al., 2020) further improves robustness by 1.0% and accuracy by 0.7% over EfficientNet-B7. We note this result can be further improved using training enhancements (Srivastava et al., 2014; Huang et al., 2016; Cubuk et al., 2019; Wong et al., 2020) (detailed in the appendices), *i.e.*, such trained EfficientNet-L1 can get an additional improvement of +1.7% for accuracy (from 80.5% to 82.2%) and +0.6% for robustness (from 58.0% to 58.6%).

Comparing to the prior art (Qin et al., 2019). Table 4 compares our best results with the prior art. With SAT, we are able to train a model with strong performance on both adversarial robustness and standard accuracy—our best model (EfficientNet-L1 + SAT) achieves 82.2% standard accuracy and 58.6% robustness, which largely outperforms the previous state-of-the-art method (Qin et al., 2019) by 9.5% on standard accuracy and 11.6% on adversarial robustness.

	Accuracy (%)	Robustness (%)
Prior art (Qin et al., 2019)	72.7	47.0
EfficientNet+SAT	**82.2 (+9.5)**	**58.6 (+11.6)**

Table 4: Comparison to the previous state-of-the-art.

Discussion. Finally, we emphasize a large reduction in the accuracy gap between adversarially trained models and standard trained models for large networks. For example, with the training setup above (with enhancements), EfficientNet-L1 achieves 84.1% accuracy in standard training, and this accuracy slightly decreases to 82.2% (-1.9%) in SAT. Note that this gap is substantially smaller than the gap in ResNet-50 of 7.1% (76.8% in standard training v.s. 69.7% in SAT). Moreover, it is also worth mentioning that the high accuracy of 82.2% provides strong support to (Ilyas et al., 2019) on arguing robust features indeed can generalize well to clean inputs.

6 CONCLUSION

In this paper, we propose smooth adversarial training, which enforces architectural smoothness via replacing non-smooth activation functions with their smooth approximations in adversarial training. SAT improves adversarial robustness without sacrificing standard accuracy or incurring additional computation cost. Extensive experiments demonstrate the general effectiveness of SAT. With EfficientNet-L1, SAT reports the state-of-the-art adversarial robustness on ImageNet, which largely outperforms the prior art (Qin et al., 2019) by 9.5% for accuracy and 11.6% for robustness.

REFERENCES

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, 2018.

Jonathan T Barron. Continuously differentiable exponential linear units. *arXiv preprint arXiv:1704.07483*, 2017.

Arjun Nitin Bhagoji, Daniel Cullina, Chawin Sitawarin, and Prateek Mittal. Enhancing robustness of machine learning systems via data transformations. In *CISS*, 2018.

Jacob Buckman, Aurko Roy, Colin Raffel, and Ian Goodfellow. Thermometer encoding: One hot way to resist adversarial examples. In *ICLR*, 2018.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *ICLR*, 2016.

Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. In *CVPR*, 2019.

Guneet S Dhillon, Kamyar Azizzadenesheli, Zachary C Lipton, Jeremy Bernstein, Jean Kossaifi, Aran Khanna, and Anima Anandkumar. Stochastic activation pruning for robust adversarial defense. In *ICLR*, 2018.

Gavin Weiguang Ding, Yash Sharma, Kry Yik Chau Lui, and Ruitong Huang. Max-margin adversarial (mma) training: Direct input space margin maximization through adversarial training. In *ICLR*, 2020.

Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 2018.

Angus Galloway, Anna Golubeva, Thomas Tanay, Medhat Moussa, and Graham W Taylor. Batch normalization is a cause of adversarial vulnerability. *arXiv preprint arXiv:1905.02161*, 2019.

Ruiqi Gao, Tianle Cai, Haochuan Li, Liwei Wang, Cho-Jui Hsieh, and Jason D Lee. Convergence of adversarial training in overparametrized networks. In *NeurIPS*, 2019.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens van der Maaten. Countering adversarial images using input transformations. In *ICLR*, 2018.

Richard HR Hahnloser, Rahul Sarpeshkar, Misha A Mahowald, Rodney J Douglas, and H Sebastian Seung. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*, 2000.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

Elad Hoffer, Itay Hubara, and Daniel Soudry. Train longer, generalize better: closing the generalization gap in large batch training of neural networks. In *Advances in Neural Information Processing Systems*, 2017.

Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *ECCV*, 2016.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *NeurIPS*, 2019.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. In *ICLR*, 2017.

Fangzhou Liao, Ming Liang, Yinpeng Dong, and Tianyu Pang. Defense against adversarial attacks using high-level representation guided denoiser. In *CVPR*, 2018.

Xuanqing Liu, Minhao Cheng, Huan Zhang, and Cho-Jui Hsieh. Towards robust neural networks via random self-ensemble. In *ECCV*, 2018.

Vishnu Suresh Lokhande, Songwong Tasneeyapant, Abhay Venkatesh, Sathya N. Ravi, and Vikas Singh. Generating accurate pseudo-labels in semi-supervised learning and avoiding overconfident predictions via hermite polynomial activations. In *CVPR*, 2020.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.

Dongyu Meng and Hao Chen. Magnet: a two-pronged defense against adversarial examples. In *CCS*, 2017.

Diganta Misra. Mish: A self regularized non-monotonic neural activation function. *arXiv preprint arXiv:1908.08681*, 2019.

Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.

Preetum Nakkiran. Adversarial robustness may be at odds with simplicity. *arXiv preprint arXiv:1901.00532*, 2019.

Tianyu Pang, Kun Xu, Chao Du, Ning Chen, and Jun Zhu. Improving adversarial robustness via promoting ensemble diversity. In *ICML*, 2019.

Tianyu Pang, Kun Xu, and Jun Zhu. Mixup inference: Better exploiting mixup to defend adversarial attacks. In *ICLR*, 2020.

Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *SP*, 2016.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *AsiaCCS*, 2017.

Aaditya Prakash, Nick Moran, Solomon Garber, Antonella DiLillo, and James Storer. Deflecting adversarial attacks with pixel deflection. In *CVPR*, 2018.

Chongli Qin, James Martens, Sven Gowal, Dilip Krishnan, Krishnamurthy Dvijotham, Alhussein Fawzi, Soham De, Robert Stanforth, and Pushmeet Kohli. Adversarial robustness through local linearization. In *NeurIPS*, 2019.

Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017.

Andras Rozsa and Terrance E Boult. Improved adversarial robustness by reducing open space risk via tent activations. *arXiv preprint arXiv:1908.02435*, 2019.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015.

Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. In *ICLR*, 2018.

Lukas Schott, Jonas Rauber, Matthias Bethge, and Wieland Brendel. Towards the first adversarially robust neural network model on mnist. In *ICLR*, 2019.

Ali Shafahi, Mahyar Najibi, Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! In *NeurIPS*, 2019.

Aman Sinha, Hongseok Namkoong, and John Duchi. Certifying some distributional robustness with principled adversarial training. In *ICLR*, 2018.

Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. In *ICLR*, 2018.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 2014.

Dong Su, Huan Zhang, Hongge Chen, Jinfeng Yi, Pin-Yu Chen, and Yupeng Gao. Is robustness the cost of accuracy?–a comprehensive study on the robustness of 18 deep image classification models. In *ECCV*, 2018.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *ICLR*, 2014.

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, 2019.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. There is no free lunch in adversarial robustness (but there are unexpected benefits). In *ICLR*, 2019.

Siyue Wang, Xiao Wang, Pu Zhao, Wujie Wen, David Kaeli, Peter Chin, and Xue Lin. Defensive dropout for hardening deep neural networks under adversarial attacks. In *ICCAD*, 2018.

Yisen Wang, Xingjun Ma, James Bailey, Jinfeng Yi, Bowen Zhou, and Quanquan Gu. On the convergence and robustness of adversarial training. In *ICML*, 2019.

Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *ICLR*, 2020.

Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: Revisiting adversarial training. In *ICLR*, 2020.

Chang Xiao and Changxi Zheng. One man's trash is another man's treasure: Resisting adversarial examples by adversarial examples. In *CVPR*, 2020.

Cihang Xie and Alan Yuille. Intriguing properties of adversarial training at scale. In *ICLR*, 2020.

Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. In *ICLR*, 2018.

Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *CVPR*, 2019.

Qizhe Xie, Eduard Hovy, Minh-Thang Luong, and Quoc Le. Self-training with noisy student improves imagenet classification. In *CVPR*, 2020.

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017.

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 *ICML*, 2019.

Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. In *ICLR*, 2016.