Parameterizing Activation Functions for Adversarial Robustness

Sihui Dai
Electrical and Computer Engineering
Princeton University
Princeton, USA
sihuid@princeton.edu

Saeed Mahloujifar
Electrical and Computer Engineering
Princeton University
Princeton, USA
sfar@princeton.edu

Prateek Mittal
Electrical and Computer Engineering
Princeton University
Princeton, USA
pmittal@princeton.edu

2022 IEEE Security and Privacy Workshops (SPW) | 978-1-6654-9643-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/SPW54247.2022.9833884

***Abstract*—Deep neural networks are known to be vulnerable to adversarially perturbed inputs. A commonly used defense is adversarial training, whose performance is influenced by model architecture. While previous works have studied the impact of varying model width and depth on robustness, the impact of using *learnable* parametric activation functions (PAFs) has not been studied. We study how using learnable PAFs can improve robustness in conjunction with adversarial training. We first ask the question: *Can changing activation function shape improve robustness?* To address this, we choose a set of PAFs with parameters that allow us to independently control behavior on negative inputs, inputs near zero, and positive inputs. Using these PAFs, we train models using adversarial training with fixed PAF shape parameter values. We find that all regions of PAF shape influence the robustness of obtained models, however only variation in certain regions (inputs near zero, positive inputs) can improve robustness over ReLU. We then combine learnable PAFs with adversarial training and analyze robust performance. We find that choice of activation function can significantly impact the robustness of the trained model. We find that only certain PAFs, such as smooth PAFs, are able to improve robustness significantly over ReLU. Overall, our work puts into context the importance of activation functions in adversarially trained models.**

I. Introduction

Deep Neural Networks (DNNs) can be fooled by perceptually insignificant perturbations known as adversarial examples [1]. A commonly used approach to defend against adversarial examples is adversarial training [2, 3] which involves training models using adversarial images. Previous studies have shown that the performance of adversarial training depends on model architecture [2, 4, 5, 6]; larger models are able to fit the training set better leading to higher robust accuracy. Additionally, a few works have studied the impact of activation function shape on robustness of adversarially trained models [4, 7]; however, it is still unclear what aspects of activation function shape are important for adversarial training. We begin by asking the question:

> *How does activation function shape impact the performance of adversarially trained models?*

To address this question, we use a set of parametric activation functions (PAFs) with a parameter controlling aspects of shape such as behavior on negative inputs, behavior on positive inputs, and behavior near zero. We vary the PAF parameter and evaluate the robustness of adversarially trained models to identify properties of activation function shape that are correlated with robustness. Our findings suggest all three aspects of activation function shape can play a significant role in adversarial robustness, but only certain aspects are able to lead to robustness higher than ReLU. We find that tuning parameters controlling behavior near zero and behavior on positive inputs can improve robustness over ReLU.

We then ask the question:

> *How do learnable parametric activation functions perform when combined with adversarial training?*

We train models using learnable PAFs with adversarial training and observe the resulting robust accuracy and learned PAF shape. We find that while introducing only 1-2 parameters into the network, certain PAFs (namely smooth PAFs) can significantly improve robustness over ReLU. For instance, when trained on CIFAR-10 with an additional 6M synthetic images from a generative model (DDPM-6M), PSSiLU, a PAF that we introduce, improves robust accuracy by 2.69% over ReLU on WideResNet(WRN)-28-10 (and 4.54% over ReLU on ResNet-18) in the ℓ_∞ threat model *while adding only 2 additional parameters into the network architecture*. The WRN-28-10 model achieves 61.96% robust accuracy, making it the top performing model in its category on RobustBench [8].

In summary, our contributions are as follows

1) We explore the impact of activation function shape on the robustness of adversarially trained models through PAFs parameterized by a single parameter controlling shape. We choose a set of PAFs which allow us to vary behavior on negative inputs, inputs near zero, and positive inputs. These PAFs include both pre-existing PAFs such as PReLU and PELU as well as PAFs that we introduce (ReBLU and PReLU$^+$). Additionally, we introduce a new activation function called PSSiLU parameterized by 2 parameters which allow us to vary multiple shape properties (behavior on negative inputs and behavior near 0) simultaneously.
2) Using our set of PAFs, we manually vary the shape parameter in order to determine which shape properties are correlated with robust accuracy on adversarially trained models. We find that for negative inputs outputting

DOI 10.1109/SPW54247.2022.00014

values near 0 can improve robustness. Additionally, we find that near zero, high bounded curvature can also improve robustness.

3) We then explore the use of PAFs with *learnable* parameters and observe their impact on robustness with adversarial training. We find that while our PAFs introduce only 1-2 parameters into the entire network (all parameters of PAFs are shared across all activations), smooth PAFs are able to improve robust accuracy over ReLU and other nonparametric activation functions. In comparison, prior works demonstrate that it takes millions of parameters in width and depth to obtain the same increase in robustness [4].

II. Related Works

a) Adversarial Attacks and Adversarial Training: Previous studies have shown that modern NNs can be fooled by perturbations known as adversarial attacks, which are imperceptible to humans, but cause NNs to predict incorrectly with high confidence [1]. These attacks can be generated in a white box [2, 8, 9, 10] or black-box [11, 12, 13] manner.

Adversarial training is a defense in which adversarial images are used to train the model. The first variant of adversarial training is PGD adversarial training [2]. Since then other variants of adversarial training have been introduced to improve robust performance [14, 15] and reduce tradeoff between natural and robust accuracy[3, 16, 17]. Recent works have also explored how to improve robustness when combined with adversarial training [4, 18]. These include techniques such as using additional data [19, 20, 21], and early stopping [22]. Croce et al. [23] provide a leaderboard for ranking defenses against adversarial attacks, and currently the top defenses on this leaderboard are all based on adversarial training.

b) Importance of Model Capacity in Adversarial Training: Prior works have indicated that the performance of adversarial training depends on model capacity. Madry et al. [2] demonstrated that large model capacity is necessary for adversarial training to successfully fit the training data. Recently, Bubeck and Sellke [24] proved that nd parameters are necessary for a model to robustly fit n d-dimensional data points. These findings raise the question, if adversarial training requires high capacity models, where in the model architecture should we introduce additional parameters? In line with this question, multiple works have studied the impact of changing the capacity of DNNs by modifying width and depth on robustness [4, 5, 6]. However, the question of how introducing parameters into activation functions impacts robustness has been unexplored. We address this question by observing the performance of parametric activation functions in conjunction with adversarial training.

c) Activation Functions and Robustness: While most works on activation functions focus on improving natural accuracy [25, 26, 27, 28], there have been a few works which explore activation functions in the adversarial setting. One line of works evaluates the impact of properties such as boundedness [29], symmetry [30], data dependency [31], learnable shape [32], and quantization [33] on robustness without using adversarial training. A more closely related line of works evaluates the performance of models using various nonparametric activation functions in conjunction with adversarial training [4, 7, 34].

In contrast to prior works, we experiment with *parametric activation functions* (PAFs), allowing us to explore a wider range of activation function shapes and understand the impact of increasing model capacity through activation functions. We will first identify qualities of activation functions that are correlated robustness by observing the robustness of adversarially trained models with various activation function shape (Section III). We then perform adversarial training on architectures with learnable PAFs and analyze their potential in improving robust accuracy (Section IV).

III. Impact of Activation Function Shape on Robustness

Existing PAFs are designed for improving natural accuracy through standard training without considering robustness, leading to the question: *how should we design a PAF for improving robustness?* One challenge is that there are many shapes that an activation function can take, leading to a large design space. Since ReLU is commonly used in DNNs, we choose a set of 6 different PAFs which can take on the shape of ReLU while allowing us to vary behavior from ReLU, which we discuss in Section III-A. Additionally, we introduce a PAF which we call PSSiLU that combines behaviors in different regions. In this section, we manually vary the parameter of PAFs and measure the robustness of adversarially trained models with different activation function shape in order to understand what aspects of shape can improve robustness through adversarial training.

A. Parametric Activation Functions Used

Since ReLU is commonly used in DNN architectures, we first consider a set of PAFs with a single parameter α that are able to model the shape of ReLU, while also allowing for variation in behavior at different regimes in the input. We divide our initial set of PAFs into 3 groups: those which capture variation on negative inputs, those which capture variation for inputs of small magnitude, and those which capture variation for large positive inputs. The shapes at varied parameter values of all PAFs that will be introduced are shown in Figure 1 and Figure 2.

a) Variation on negative inputs: To capture variation on negative inputs, we consider parametric ReLU (PReLU) [28] and parametric ELU (PELU) [25] defined as follows:

$$\text{PReLU}_\alpha(x) = \begin{cases} \alpha x & x \leq 0 \\ x & x > 0 \end{cases}$$

$$\text{PELU}_\alpha(x) = \begin{cases} \alpha(e^x - 1) & x \leq 0 \\ x & x > 0 \end{cases}$$

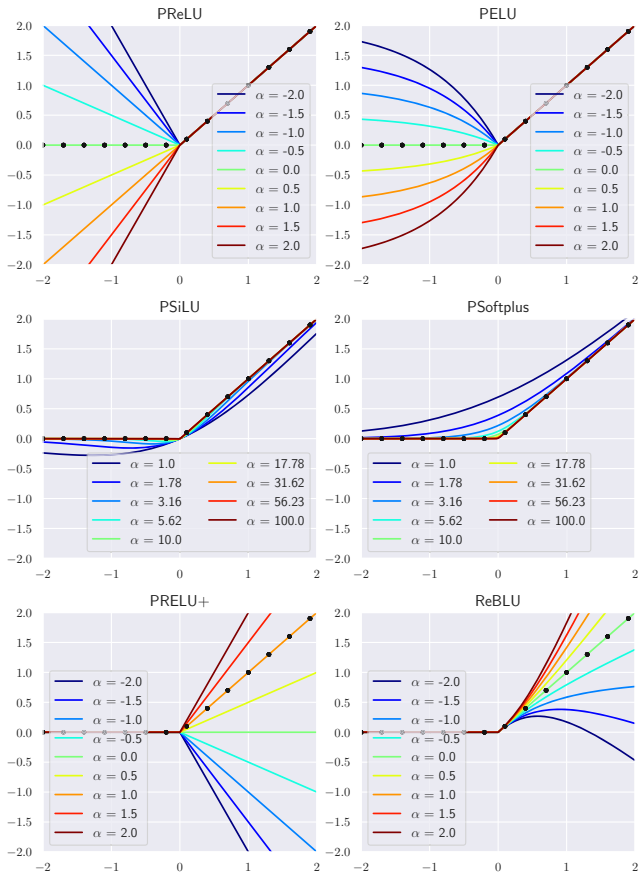

Fig. 1. Visualization of parametric activation functions at various values of parameter α.

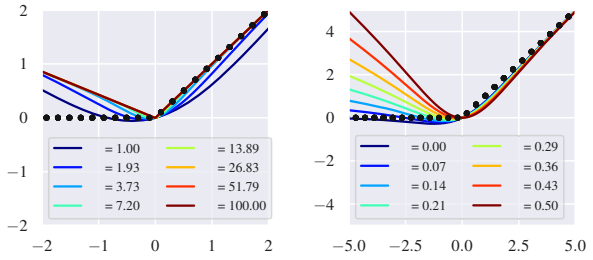

Fig. 2. Shape of PSSiLU at various values of α and β. Left: β is fixed to 0.3 while α is varied. Right: α is fixed to 1 while β is varied. ReLU is given by the dotted black line. We can see that α controls the curvature of the function near 0 while β controls the behavior on negative inputs.

b) Variation near zero: To capture variation for inputs near zero, we consider two parametric activation functions parametric SiLU (PSiLU) [27] and parametric Softplus (PSoftplus) [35]. These activation functions are defined as follows:

$$\text{PSiLU}_\alpha(x) = x\sigma(\alpha x) \qquad \text{PSoftplus}_\alpha = \frac{1}{\alpha}\log(1+e^{\alpha x})$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function. For both PAFs, the parameter α controls its curvature, the maximum value of the second derivative. We can see that at large values of α, both PSiLU and PSoftplus approach the shape of ReLU. Unlike PELU and PReLU, PSiLU and PSoftplus also have the property of being smooth, which prior work [7] suggests may improve the performance of adversarial training.

c) Combining properties: We introduce a PAF, which we call PSSiLU (parametric shifted SiLU) which allows for both variation along negative inputs and variation near zero via 2 parameters α and β:

$$\text{PSSiLU}_{\alpha,\beta}(x) = x(\sigma(\alpha x) - \beta)/(1-\beta) \tag{1}$$

where $\alpha, \beta > 0$, $\beta < 1$, and $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function. At $\beta = 0$, PSSiLU's behavior matches that of PSiLU. The impact of changing these parameters on the shape of PSSiLU is shown in Figure 2. α controls curvature around 0 while β controls behavior on negative inputs. Increasing β allows PSSiLU's output on input $x < 0$ to grow with the magnitude of x similar to PReLU. Similar to PSiLU and PSoftplus, PSSiLU is smooth.

d) Variation on positive inputs: To capture variation on positive inputs, we introduce two PAFs: one which we call Positive PReLU (PReLU^+) and the other which we call Rectified BLU (ReBLU). PReLU^+ has a parameter controlling the slope of the linear portion of ReLU and is defined as:

$$\text{PReLU}^+_\alpha(x) = \begin{cases} 0 & x \le 0 \\ \alpha x & x > 0 \end{cases}$$

We note that the function class modelled by PReLU^+ is the same as the function class modelled by ReLU since for every PReLU^+ network, you can construct a corresponding ReLU network by scaling the weight parameters. Thus, any difference in performance between PReLU^+ and ReLU is due to optimization.

Unlike PReLU^+, ReBLU allows for nonlinear behavior on positive inputs and is based off Bendable Linear Unit (BLU) defined as $\text{BLU}_\alpha = \alpha(\sqrt{x^2+1}-1)+x$ [36]. To allow BLU to take the shape of ReLU for comparison, we modify BLU so that it is piecewise and outputs 0 for all negative inputs. We define ReBLU as follows:

$$\text{ReBLU}_\alpha(x) = \begin{cases} 0 & x \le 0 \\ \text{BLU}_\alpha(x) & x > 0 \end{cases}$$

B. Identifying Properties of Shape Correlated with Robustness on Adversarially Trained Models

Using our set of 6 PAFs with a single parameter, we vary activation function shape and measure the change in robustness in order to determine which properties of activation shape are correlated with robustness of adversarially trained models.

a) Experimental Setup: We train ResNet-18 models on CIFAR-10 with 10 step PGD adversarial training [2]. We use an ℓ_∞ adversary with radius $\epsilon = \frac{8}{255}$ and step size $\frac{2}{255}$. For optimization, we use SGD with initial learning rate of 0.1 and train for a total of 200 epochs. We decrease the learning rate

by a factor of 10 at the 100th and 150th epoch. For each PAF, we train a model with parameter α fixed to the values shown in Figure 1. We measure the robustness of trained models using AutoAttack [8].

We report the measured AutoAttack robust accuracy and clean accuracy of models for different PAF parameter values in Figure 3.

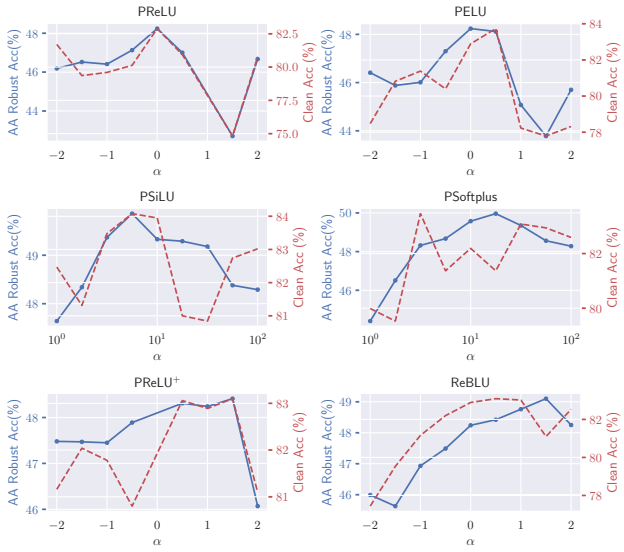

Fig. 3. Square robust accuracy and average minimum PGD radius for ResNet-18 models trained on CIFAR-10 with various parameter α. Results are computed over 3 trials. Red points indicate the clean accuracy while blue points indicate the AutoAttack robust accuracy across models.

b) Robustness is highest when behavior on negative values is near 0: Across both PReLU and PELU, varying the parameter α can lead to high variation in robust performance. We find that robust accuracy ranges from ~44% to 48% across PReLU and PELU, which suggests that this behavior on negative inputs is correlated with robustness. We find that when $\alpha = 0$, we achieve the highest robust accuracy. At $\alpha = 0$, both PAFs take the shape of ReLU, suggesting that ReLU has optimal behavior on negative inputs.

c) High bounded curvature is correlated with robustness: Across PSiLU and PSoftplus, for which α controls curvature, we find that the optimal performance is obtained at $\alpha > 1$. $\alpha = 1$ is the parameter setting corresponding to SiLU and Softplus activations used in practice, and we find that these activations can perform worse than ReLU. From our plot, SiLU obtains ~47.5% robust accuracy, Softplus obtains ~44% robust accuracy. In comparison, ReLU (shown at $\alpha = 0$ on the plot for PReLU) obtains ~48% robust accuracy. This suggests that while previous work [7] demonstrates that smooth activation functions can improve robustness through adversarial training, the function class that can be modelled by these activation functions is also important. We find that high curvature, which brings the shape of PSiLU and PSoftplus closer to the shape of ReLU, can improve robust accuracy. We also find that for both activation functions, the robustness decreases when α becomes large. For instance for ReLU which has infinite curvature, we can obtain only ~48% accuracy while the maximum accuracy of PSiLU and PSoftplus is above 49%.

d) On ReBLU, superlinear behavior is correlated with robustness: We find that for ReBLU, increasing parameter α improves robustness. On this PAF, $\alpha = 0$ corresponds to the performance of ReLU. From Figure 1, $\alpha > 0$ corresponds to superlinear growth in the positive portion of ReBLU, and these values of α can improve over the performance of ReLU. This result suggests that superlinear behavior may be correlated with robustness in adversarially trained models. We do not observe the same trend for PReLU$^+$ for which there is generally less variation in robustness across α due to the fact that PReLU$^+$ captures the same function class as ReLU.

Overall, our findings suggest that the choice of activation function shape can improve or detriment the robust accuracy of PAFs. Additionally, our findings suggest that certain activation function shapes such as those with high bounded curvature and ReBLU with superlinear growth on positive inputs can improve robustness over a corresponding ReLU network.

IV. Investigating the Performance of Adversarially Trained Models Using Parametric Activation Functions

We now combine learnable PAFs with adversarial training to investigate the impact of incorporating parameters into activation functions on adversarial training. Specifically, we add α (and β for PSSiLU) to the parameter set θ that we optimize during adversarial training. We share PAF parameters across all layers in the network, so that PSSiLU only introduces two additional parameters into the model while all other PAFs introduce one new parameter. We also train models using the commonly used nonparametric activation functions: ReLU, ELU, SiLU, Softplus. ELU, SiLU, and Softplus correspond to $\alpha = 1$ for PELU, PSiLU, and PSoftplus respectively.

We perform experiments on WRN-28-10 and ResNet-18 architectures on CIFAR-10. We also experiment with using additional data during training. For additional CIFAR-10 data, we use DDPM-6M [37], a set of 6M CIFAR-10 images generated by DDPM, a generative model [38] which have been shown to improve the robustness of adversarially trained models [20, 21], and labelled by a 98.5% accurate BiT model [39]. For the bulk of our experiments, we use 10-step PGD adversarial training [2] and focus on ℓ_∞ attacks. We present results on ResNet-18 in Table I and results on WRN-28-10 in Table II.

a) Not all parameters are equal: We find that although networks using PAFs capture a larger function class than ReLU networks, not all PAFs can obtain robust accuracy higher than ReLU. For instance, across both Table I and Table II, we find that PReLU and PELU consistently perform *worse* than ReLU despite both being able to capture the shape of ReLU. This suggests that there may be some difficulty in optimization for these activation functions which prevent them from learning a more optimal α parameter value of 0.

	CIFAR-10		+DDPM-6M	
Activation	Natural	AA	Natural	AA
ReLU	82.84	48.46	82.83	53.67
PReLU	83.05	47.27	83.27	53.66
ELU	80.47	45.43	82.47	51.59
PELU	82.51	47.34	83.07	53.29
Softplus	80.46	44.64	79.44	49.41
PSoftplus	83.74	49.28	84.56	56.78
PReLU$^+$	81.96	47.62	83.91	54.09
ReBLU	83.15	48.22	83.63	54.21
SiLU	83.80	47.41	83.53	54.07
PSiLU	83.96	**49.64**	84.73	55.20
PSSiLU	**84.10**	49.27	**84.79**	58.21

TABLE I
NATURAL AND ROBUST ACCURACY OF PGD ADVERSARIALLY TRAINED RESNET-18 MODELS OF VARIOUS ACTIVATION FUNCTIONS ON CIFAR-10 WITH RESPECT TO ℓ_∞ ATTACKS WITH RADIUS 0.031. THE AA COLUMN GIVES THE ROBUST ACCURACY OF ATTACKS GENERATED THROUGH AUTOATTACK ON THE TEST SET. WE HIGHLIGHT ROBUST ACCURACIES LARGER THAN RELU IN PURPLE.

	CIFAR-10		+DDPM-6M	
Activation	Natural	AA	Natural	AA
ReLU	86.29	51.95	85.92	59.27
PReLU	86.88	48.51	86.04	58.74
ELU	77.67	43.90	81.09	50.79
PELU	86.89	48.41	85.83	58.90
Softplus	79.99	44.41	78.86	49.14
PSoftplus	86.99	**52.74**	86.60	60.94
PReLU$^+$	**87.18**	45.05	86.05	59.13
ReBLU	86.80	52.25	86.39	59.62
SiLU	83.95	48.39	84.90	55.10
PSiLU	87.13	51.92	86.47	60.37
PSSiLU	86.41	51.47	**87.02**	**61.96**

TABLE II
NATURAL AND ROBUST ACCURACY OF PGD ADVERSARIALLY TRAINED WRN-28-10 MODELS OF VARIOUS ACTIVATION FUNCTIONS ON CIFAR-10 WITH RESPECT TO ℓ_∞ ATTACKS WITH RADIUS 0.031. THE AA COLUMN GIVES THE ROBUST ACCURACY OF ATTACKS GENERATED THROUGH AUTOATTACK ON THE TEST SET. WE HIGHLIGHT ROBUST ACCURACIES LARGER THAN RELU IN PURPLE.

Meanwhile, smooth PAFs including PSoftplus, PSiLU, and PSSiLU consistently achieve robust accuracy that is comparable to or higher than ReLU. For instance on PSSiLU on WRN-28-10 improves over the performance of ReLU by 2.28% without extra data and 2.69% with additional data. The importance of smoothness was studied by Xie et al. [7] and [4]. Xie et al. [7] found that smooth activation functions can improve the performance of adversarial training on ImageNet while Gowal et al. [4] were unable to observe the same trend in CIFAR-10. We find that when smooth nonparametric activation functions are unable to outperform ReLU, their parametric counterparts are able to, suggesting that the combination of smoothness and the flexibility of our PAFs to model ReLU improves robustness.

Additionally, despite not being smooth, ReBLU can also consistently achieve robustness on par with or higher than ReLU. However, this improvement over ReLU is quite small; for instance ReBLU only achieves a 0.35% improvement over ReLU on WRN-28-10 with additional DDPM-6M data.

Overall, we find that the importance of parameters are generally consistent with our results in Section III, where we found that when manually varying shape paramete PELU and PReLU are optimal at ReLU's shape while for PSiLU, PSoftplus, and ReBLU, we are able to find a parameter setting that led to higher robustness than ReLU.

b) By adding only two additional parameters, PSSiLU can significantly improve robust accuracy over ReLU.: We observe that for ResNet-18 and WRN-28-10, PSSiLU achieves both high clean and high robust accuracy. Compared to ReLU, we observe that PSSiLU improves robust performance by a total of 4.54% *while only adding 2 parameters into the network architecture*. Moreover, with the additional DDPM-6M data on ResNet-18, PSSiLU improves over the robust performance of SiLU by 4.14% and PSiLU by 3.01%, both of which can be modeled by PSSiLU.

On WRN-28-10, PSSiLU achieves 87.02% clean accuracy and 61.96% robust accuracy, improving on clean accuracy by 1.10% and robust accuracy by 2.69% over ReLU, making our WRN-28-10 model the best performing in its category on RobustBench [8]. This improvement in robust accuracy is significant; prior works have shown that it takes millions of additional parameters through varying width and depth of CNNs in order to achieve a 1-2% increase in robustness on WRN on CIFAR-10 [4].

The improvement of PSSiLU over ReLU demonstrates the potential of using learnable activation functions in conjunction with adversarial training. Additionally, the significance of the improvement in robust accuracy over ReLU emphasizes the importance of activation function shape.

In summary, we find that even when parameters are learnable, we cannot always improve robustness over ReLU with PAFs. We find that certain PAFs, specifically smooth PAFs and ReBLU are able to improve robustness over ReLU, while other PAFs such as PReLU and PELU are always suboptimal, despite being able to capture the function class of ReLU networks. This result demonstrates that optimization may play a role in performance of PAFs. Additionally, we find that differences in robustness between models of different activation function can be significant; for instance, PSSiLU improves on robust accuracy by 2.69% over ReLU when trained with additional DDPM data. We now move on to visualize the learned shapes of these PAFs.

A. Visualizing Learned Shapes of Parametric Activation Functions

Previously, in Section III, we manually varied the shape parameter of PAFs in order to understand how behavior in different regions of input is correlated with robustness. We present the learned shapes of the 6 PAFs with single parameter in Figure 4 for models trained without additional data.

We find the shapes of learned activation functions are consistent with our analysis in Section III across architectures. For instance, in Section III, we found that for PSiLU and PSoftplus, large values of α (higher curvature) leads to

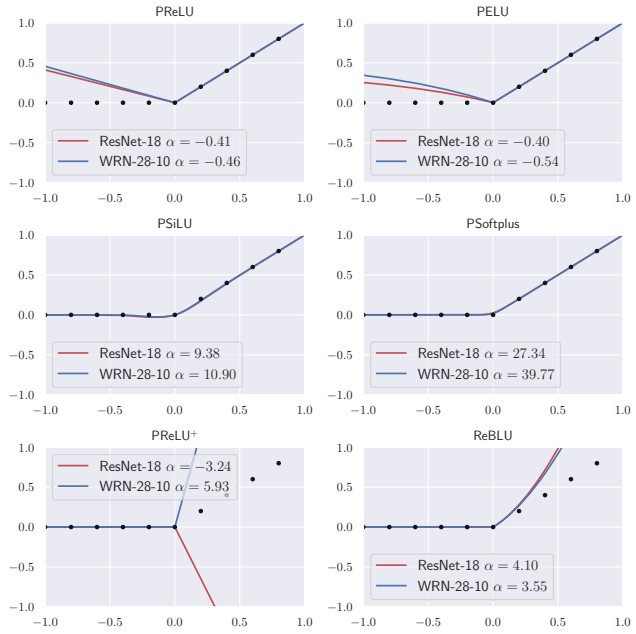

Fig. 4. Learned shapes of PAFs across models trained on CIFAR-10. Red lines indicate activation shape for the ResNet-18 model while blue lies indicate activation shape for the WRN-28-10 model.

improved robustness. We find that when the parameter is learnable, the best performing model also optimizes towards these larger values of α. Similarily, we find that for ReBLU, the shape learned has superlinear behavior which we found is correlated with higher robustness in Section III. However, we find that while trends are conserved, the values for α learned do not exactly coincide with the values of the α with optimal robustness in Figure 3. For instance, in Figure 3, we find that the optimal value of α is ~5 when training with a fixed activation value. However, we find that when training with a learnable parameter, α tends towards ~10.

Additionally, the relation to trends from Section III also help to explain why PReLU and PELU perform suboptimally compared to ReLU. We find that for PReLU and PELU, the activation function shape optimizes towards small negative values of α while from Section III, we found that the behavior of these activation functions is optimal when PReLU and PELU have shape near that of ReLU ($\alpha = 0$).

V. Limitations and Future Directions

In our work, we showed that the choice of activation function is important to robust accuracy obtained through adversarial training. We identified aspects of activation function behavior which allow activation functions to improve robustness over ReLU. These include outputting values near zero on negative inputs, having high bounded curvature, and superlinear growth on positive inputs (in the case of ReBLU). We now suggest several future directions in order to address limitations of this work.

a) Understanding why certain properties of activation function shape improve robustness: While we demonstrated that activation function shape impacts robustness, we do not have explanations for this phenomenon. In a future direction, we would like to understand why certain activation function shapes are more optimal/suboptimal than others when used with adversarial training. For instance, we found that PReLU and PELU were unable to improve over ReLU, leading to the question: why does changing behavior on negative inputs degrade robust accuracy? Additionally, we found that interestingly ReBLU, which is not smooth, can improve robust accuracy over ReLU when ReBLU has superlinear behavior on positive inputs. In the future, we would like to further examine why this occurs.

b) Combining properties of activation function shape: In this work, we introduced PSSiLU which allows us to vary behavior on both negative inputs and behavior near zero (while having the nice property of being smooth). To further investigate the impact of activation function shape, it would be good to also introduce PAFs which allow for other combinations of activation shape properties. For instance, since ReBLU can improve robustness over ReLU, we would like to combine ReBLU with smooth PAFs controlling curvature such as PSiLU and PSoftplus to see if we can further improve on robust accuracy.

VI. Conclusion

In this work, we studied the impact activation function shape on robustness through adversarial training. We find that not all parameterizations of activation functions are able to improve robustness over ReLU, but find that smooth activation functions with a parameter controlling curvature and a PAF we introduce named ReBLU are able to improve robustness over ReLU. We combine learnable PAFs with adversarial training and find that by introducing as many as 1-2 additional parameters into the network architecture, PAFs can significantly improve robustness over ReLU. Overall, this work demonstrates the importance of activation functions in adversarial training and the potential of PAFs for enhancing robustness of machine learning against adversarial examples.

Acknowledgements

We would like to thank Vikash Sehwag and Chong Xiang for their discussions on this project and feedback on the paper draft. This work was supported in part by the National Science Foundation under grants CNS-1553437 and CNS-1704105, the ARL's Army Artificial Intelligence Innovation Institute (A2I2), the Office of Naval Research Young Investigator Award, the Army Re- search Office Young Investigator Prize, Schmidt DataX award, and Princeton E-ffiliates Award. This material is also based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-2039656. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6199.

[2] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

[3] 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 Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 7472–7482. PMLR, 2019. URL http://proceedings.mlr.press/v97/zhang19p.html

[4] 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.

[5] Cihang Xie and Alan Yuille. Intriguing properties of adversarial training at scale. In *International Conference on Learning Representations*, 2019.

[6] Boxi Wu, Jinghui Chen, Deng Cai, Xiaofei He, and Quanquan Gu. Does network width really help adversarial robustness? *arXiv preprint arXiv:2010.01279*, 2020.

[7] Cihang Xie, Mingxing Tan, Boqing Gong, Alan Yuille, and Quoc V Le. Smooth adversarial training. *arXiv preprint arXiv:2006.14536*, 2020.

[8] 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.

[9] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6572.

[10] 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.

[11] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *International Conference on Learning Representations*, 2018.

[12] 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.

[13] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*, 2016.

[14] Yisen Wang, Xingjun Ma, James Bailey, Jinfeng Yi, Bowen Zhou, and Quanquan Gu. On the convergence and robustness of adversarial training. In *International Conference on Machine Learning*, 2019.

[15] Jingfeng Zhang, Jianing Zhu, Gang Niu, Bo Han, Masashi Sugiyama, and Mohan Kankanhalli. Geometry-aware instance-reweighted adversarial training. In *International Conference on Learning Representations*, 2020.

[16] Jingfeng Zhang, Xilie Xu, Bo Han, Gang Niu, Lizhen Cui, Masashi Sugiyama, and Mohan Kankanhalli. Attacks which do not kill training make adversarial learning stronger. In *ICML*, 2020.

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

[18] Tianyu Pang, Xiao Yang, Yinpeng Dong, Hang Su, and Jun Zhu. Bag of tricks for adversarial training. In *International Conference on Learning Representations*, 2020.

[19] Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C Duchi, and Percy S Liang. Unlabeled data improves adversarial robustness. In *Advances in Neural Information Processing Systems*, pages 11192–11203, 2019.

[20] Vikash Sehwag, Saeed Mahloujifar, Tinashe Handina, Sihui Dai, Chong Xiang, Mung Chiang, and Prateek Mittal. Improving adversarial robustness using proxy distributions. *arXiv preprint arXiv:2104.09425*, 2021.

[21] Sylvestre-Alvise Rebuffi, Sven Gowal, Dan A. Calian, Florian Stimberg, Olivia Wiles, and Timothy A. Mann. Fixing data augmentation to improve adversarial robustness. *CoRR*, abs/2103.01946, 2021. URL https://arxiv.org/abs/2103.01946.

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

[23] Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. *arXiv preprint arXiv:2010.09670*, 2020.

[24] Sébastien Bubeck and Mark Sellke. A universal law of robustness via isoperimetry. *arXiv preprint arXiv:2105.12806*, 2021.

[25] Djork-Arné Clevert, Thomas Unterthiner, and Sepp

Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL http://arxiv.org/abs/1511.07289.

[26] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323. JMLR Workshop and Conference Proceedings, 2011.

[27] Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Searching for activation functions. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Workshop Track Proceedings*, 2018. URL https://openreview.net/forum?id=Hkuq2EkPf.

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[29] Valentina Zantedeschi, Maria-Irina Nicolae, and Ambrish Rawat. Efficient defenses against adversarial attacks. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 39–49, 2017.

[30] Qiyang Zhao and Lewis D. Griffin. Suppressing the unusual: towards robust cnns using symmetric activation functions. *CoRR*, abs/1603.05145, 2016. URL http://arxiv.org/abs/1603.05145.

[31] Bao Wang, Alex T Lin, Zuoqiang Shi, Wei Zhu, Penghang Yin, Andrea L Bertozzi, and Stanley J Osher. Adversarial defense via data dependent activation function and total variation minimization. *arXiv preprint arXiv:1809.08516*, 2018.

[32] Mohammadamin Tavakoli, Forest Agostinelli, and Pierre Baldi. Splash: Learnable activation functions for improving accuracy and adversarial robustness. *arXiv preprint arXiv:2006.08947*, 2020.

[33] Adnan Siraj Rakin, Jinfeng Yi, Boqing Gong, and Deliang Fan. Defend deep neural networks against adversarial examples via fixed and dynamic quantized activation functions. *arXiv preprint arXiv:1807.06714*, 2018.

[34] Vasu Singla, Sahil Singla, David Jacobs, and Soheil Feizi. Low curvature activations reduce overfitting in adversarial training. *arXiv preprint arXiv:2102.07861*, 2021.

[35] Charles Dugas, Yoshua Bengio, François Bélisle, Claude Nadeau, and René Garcia. Incorporating second-order functional knowledge for better option pricing. *Advances in neural information processing systems*, pages 472–478, 2001.

[36] Luke B. Godfrey. An evaluation of parametric activation functions for deep learning. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pages 3006–3011, 2019. doi: 10.1109/SMC.2019.8913972.

[37] Preetum Nakkiran, Behnam Neyshabur, and Hanie Sedghi. The deep bootstrap framework: Good online learners are good offline generalizers. In *International Conference on Learning Representations*, 2020.

[38] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/

[39] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In *European conference on computer vision*, pages 491–507. Springer, 2020.