

PARAMETERIZING ACTIVATION FUNCTIONS FOR ADVERSARIAL ROBUSTNESS

Anonymous authors

Paper under double-blind review

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 capacity. While previous works have studied the impact of varying model width and depth on robustness, the impact of increasing capacity by 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: *how should we incorporate parameters into activation functions to improve robustness?* To address this, we analyze trends between activation shape and robustness of standard trained models. We find that behavior on negative inputs and curvature are correlated with robustness against a weak adversary. We then introduce a new PAF, which we call Parametric Shifted Sigmoidal Linear Unit (PSSiLU), whose parameters control these properties. We combine learnable PAFs (including PReLU, PSoftplus and PSSiLU) with adversarial training and analyze robust performance. We find that PAFs optimize towards trends observed on standard trained models, suggesting that behavior on negative inputs and curvature are important for allowing the model to better fit the training data. Additionally, we find that while introducing only 1-2 learnable parameters into the network, smooth PAFs can significantly increase robustness over ReLU. For instance, when trained on CIFAR-10 with additional synthetic data, PSSiLU improves robust accuracy by 4.54% over ReLU on ResNet-18 and 2.69% over ReLU on WRN-28-10 in the ℓ_∞ threat model while adding only 2 additional parameters into the network architecture. The PSSiLU WRN-28-10 model achieves 61.96% AutoAttack accuracy, improving over the state-of-the-art robust accuracy on RobustBench (Croce et al., 2020). Overall, our work puts into context the importance of activation functions in adversarially trained models.

1 INTRODUCTION

Deep Neural Networks (DNNs) can be fooled by perceptually insignificant perturbations known as adversarial examples (Szegedy et al., 2014). A commonly used approach to defend against adversarial examples is adversarial training (Madry et al., 2018; Zhang et al., 2019) which involves training models using adversarial images. Previous studies have shown that the performance of adversarial training depends on model capacity (Madry et al., 2018); larger models are able to fit the training set better leading to higher robust accuracy. These findings raise the question, if adversarial training requires high capacity models, where in the model architecture should we introduce additional parameters? Many studies have observed the impact of factors such as model width and depth (Gowal et al., 2020; Wu et al., 2020a; Xie & Yuille, 2019), but to the best of our knowledge, the potential of increasing capacity through learnable parametric activation functions (PAFs) has not been studied. We first ask the question

How should we parameterize activation functions to improve robustness?

To find the right way to parameterize activation functions for robustness, we first need to identify which aspects of activation function shape impact robustness. 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 standard trained models to identify properties of activation function shape that are correlated with robustness. Using standard trained models allows us to decouple the impact of

activation functions from the impact of adversarial training. Surprisingly, we observe a clear trend between activation function shape and robustness **against a weak adversary** of standard trained models. We find that we can increase robustness by adjusting ReLU to output positive values on negative inputs and to have high finite curvature, the maximum value of the second derivative. We combine these properties into a new PAF which we call Parametric Shifted Sigmoidal Linear Unit (PSSiLU) shown in Figure 1. PSSiLU uses two parameters α and β ; β controls the behavior on negative inputs and α controls curvature. We then ask the question:

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

We train models using PAFs (including PReLU, PSoftplus, and PSSiLU) with learnable parameter with adversarial training and observe the resulting AutoAttack robust accuracy (Croce & Hein, 2020) and learned shape of the activation function. We find that while introducing only 1-2 parameters into the network, certain PAFs can significantly improve robustness over ReLU (Table 1). For instance, when trained on CIFAR-10 with an additional 6M synthetic images from a generative model (DDPM-6M), PSSiLU 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 (Croce & Hein, 2020). We find that PAFs can increase robustness in two ways: 1) through increasing expressivity, allowing the model to better fit the training data, and 2) through optimization, allowing the model to reach a more optimal minimum. Additionally, we find that activation functions optimize towards the properties observed to increase robustness on standard trained models, suggesting that these properties also allow the model to better fit the data.

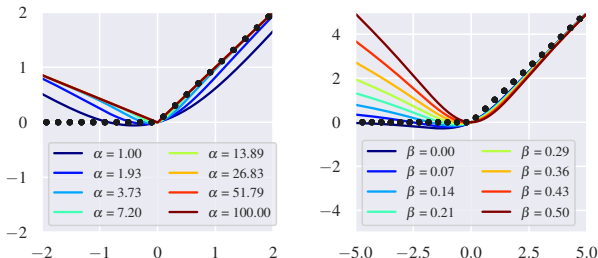


Figure 1: Shape of PSSiLU at various values of α and β . Left: β is fixed to 0.3 while α is varied. Right: α is fixed to 1 while β is varied. ReLU is given by the dotted black line. We can see that α controls the curvature of the function near 0 while β controls the behavior on negative inputs.

	Activation	AA
Non	Softplus	49.14
	SiLU	55.10
	ReLU	59.27
PAF	PSoftplus	60.94
	PSSiLU	60.37
	PSSiLU	61.96

Table 1: AutoAttack robust accuracy of WRN-28-10 models trained using PGD adversarial training on CIFAR-10+DDPM-6M. Nonparametric activation functions are labeled by “Non”. Parametric activation functions are labeled by “PAF”.

In summary, our contributions are as follows

1. We find that in the absence of adversarial training, certain properties of activation function shape (namely positive outputs on negative inputs and high finite curvature) are correlated with robustness of standard trained models against a weak adversary. Using these observations, we introduce a new PAF which we call PSSiLU with two parameters which control these properties (Figure 1).
2. While prior works only explore the use of activation functions with fixed parameter with adversarial training, we explore the use of PAFs with *learnable* parameter and observe their impact on robustness with adversarial training. We find that PAFs optimize towards the same properties which improve robustness in the standard training setting, suggesting that these properties also allow the model to better fit the training data.
3. We unlock the full potential of using learnable PAFs with adversarial training by training with additional synthetic data, increasing robust accuracy over nonparametric activation functions (Table 1). The PAFs tested only add 1-2 parameters into the entire network (all parameters of PAFs are shared across all activations), but we find that smooth PAFs are able to improve robust accuracy over ReLU and other nonparametric activation functions. This emphasizes the importance of considering activation functions in adversarial training.

4. We find that when trained on CIFAR-10 with additional synthetic data, PSSiLU improves by 2.69% over ReLU on WRN-28-10 (and 4.54% over ReLU on ResNet-18) in the ℓ_∞ threat model, making it the top performing model in its category on RobustBench (Croce et al., 2020). Additionally, we find that the family of activation functions captured by PSSiLU consistently achieves high robust accuracy, outperforming RELU across multiple datasets, architectures, perturbation types, and sources of additional data.

2 RELATED WORKS

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 (Szegedy et al., 2014). These attacks can be generated in a white box (Goodfellow et al., 2015; Madry et al., 2018; Carlini & Wagner, 2017; Croce & Hein, 2020) or black-box (Brendel et al., 2018; Andriushchenko et al., 2020; Papernot et al., 2016) 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 (Madry et al., 2018). Since then other variants of adversarial training have been introduced to improve robust performance (Wang et al., 2019; Zhang et al., 2020b) and reduce tradeoff between natural and robust accuracy (Zhang et al., 2019; 2020a; Wu et al., 2020b). Recent works have also explored how to improve robustness when combined with adversarial training (Gowal et al., 2020; Pang et al., 2020). These include techniques such as using additional data (Carmon et al., 2019; Sehwag et al., 2021; Rebuffi et al., 2021), and early stopping (Rice et al., 2020). Croce et al. (2020) provide a leaderboard for ranking defenses against adversarial attacks, and currently the top defenses on this leaderboard are all based on adversarial training.

Importance of Model Capacity in Adversarial Training. Prior works have indicated that the performance of adversarial training depends on model capacity. Madry et al. (2018) demonstrated that large model capacity is necessary for adversarial training to successfully fit the training data. Recently, Bubeck & Sellke (2021) 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 (Wu et al., 2020a; Xie & Yuille, 2019; Gowal et al., 2020). 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.

Activation Functions and Robustness. While most works on activation functions focus on improving natural accuracy (Clevert et al., 2016; Glorot et al., 2011; Ramachandran et al., 2018; He et al., 2015), 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 (Zantedeschi et al., 2017), symmetry (Zhao & Griffin, 2016), data dependency (Wang et al., 2018), learnable shape (Tavakoli et al., 2020), and quantization (Rakin et al., 2018) 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 (Xie et al., 2020; Gowal et al., 2020; Singla et al., 2021).

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 which have a direct impact on robustness by observing standard trained models in order to design a PAF to use with adversarial training (Section 3). We then combine PAFs with *learnable parameter* with adversarial training and analyze their potential in improving robust accuracy obtained through adversarial training (Section 4).

3 SEARCHING FOR A GOOD PARAMETERIZATION

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?*

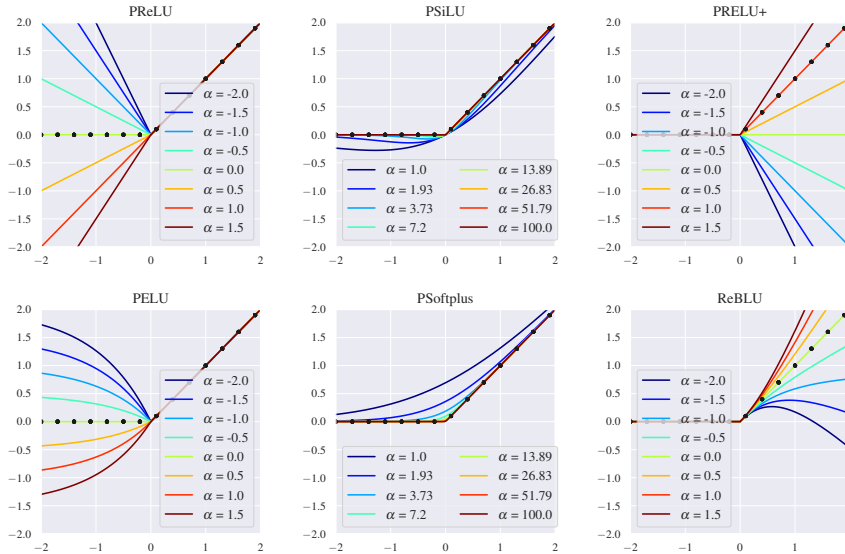


Figure 2: Visualization of parametric activation functions at various values of parameter α .

One challenge in designing PAFs for robustness 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 and has good performance, 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 3.1. By controlling the shape of these PAFs, we identify qualities of activation functions that **are correlated with** robustness.

This raises another question: *How should we measure what parameterizations are considered good?* We can divide the impact of activation functions on robustness into the direct impact of activation function shape (restriction bias) and the impact of activation functions in conjunction with optimization through adversarial training (preference bias). In this section, we focus on the restriction bias by measuring the robustness of standard trained models against a weak adversary in Section 3.2. We use a weak adversary since standard trained models are not robust against strong attacks. By observing standard trained models, we identify properties in activation function shape that **are correlated with** robustness. We then combine the observed properties into a novel PAF in Section 3.3 for use with adversarial training (Section 4).

3.1 ACTIVATION FUNCTION SEARCH SPACE

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 values of parameter α of all activation functions that will be introduced are shown in Figure 2.

To capture variation on negative inputs, we consider parametric ReLU (PReLU) (He et al., 2015) and parametric ELU (PELU) (Clevert et al., 2016) defined as follows:

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

To capture variation for inputs near zero, we consider two continuous parametric activation functions parametric SiLU (PSiLU) (Ramachandran et al., 2018) and parametric Softplus (PSoftplus) (Dugas et al., 2001). These activation functions are defined as follows:

$$\text{PSiLU}_\alpha(x) = x\sigma(\alpha x) \quad \text{PSoftplus}_\alpha = \frac{1}{\alpha} \log(1 + e^{\alpha x}) \quad (2)$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function.

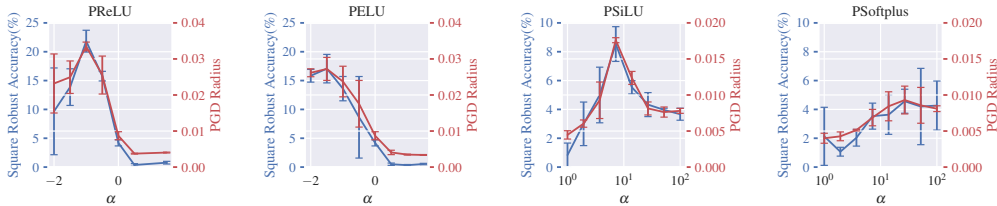


Figure 3: Square robust accuracy and average minimum PGD radius for ResNet-18 models trained on CIFAR-10 with various parameter α . Results are computed over 3 trials. Red points indicate the measured PGD radius while blue points indicate the Square robust accuracy across models.

To capture variation on positive inputs, we introduce two activation functions: 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. ReBLU allows for nonlinear behavior on positive inputs and is based off Bendable Linear Unit (BLU) defined as $BLU_{\alpha} = \alpha(\sqrt{x^2 + 1} - 1) + x$ (Godfrey, 2019). 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 PReLU⁺ and ReBLU as follows:

$$PReLU_{\alpha}^{+}(x) = \begin{cases} 0 & x \leq 0 \\ \alpha x & x > 0 \end{cases} \quad ReBLU_{\alpha}(x) = \begin{cases} 0 & x \leq 0 \\ BLU_{\alpha}(x) & x > 0 \end{cases} \quad (3)$$

3.2 IDENTIFYING PARAMETERS CONDUCIVE TO ROBUSTNESS ON STANDARD TRAINED MODELS

Using our set of 6 PAFs, we vary the shapes of activation functions and measure **the change in robustness** in order to design a new PAF which models more robust activation function shapes. To disentangle the impact of activation function shape from the impact of training, we analyze the robustness of standard trained models. We use a weak adversary in this analysis because standard trained models are not robust; using a strong adversary would make it difficult to see the impact of activation function shape. Note that we do not expect the standard trained model to achieve any robustness against strong attacks even with PAFs. However, in Section 4, we will experiment with adversarially trained models against a strong adversary.

For our weak adversary, we use both white box and black box attacks. For white box attack, we search for the smallest perturbation radius that leads to misclassification on 4-step PGD perturbed inputs. For black box attacks, we use a query-restricted black-box adversary (Square with 1000 queries (Andriushchenko et al., 2020; Croce & Hein, 2020)) and measure the robust accuracy of the models on adversarial examples. Additional details on experimental setup are located in Appendix A.1. We present the measured Square robust accuracy and PGD radius for ResNet-18 models trained on CIFAR-10 at various parameter α in Figure 3.

From Figure 3, we observe clear trends across PReLU, PELU, PSiLU, and PSoftplus and find that Square robust accuracy and PGD radius are highly correlated, suggesting that these trends are not the result of obfuscated gradients (Athalye et al., 2018). Additionally, we found that these trends generalize to other model architectures and datasets. We provide results for ResNet-18 models on CIFAR-100 and ImageNette datasets and results for WRN-28-10 and VGG-16 models trained on CIFAR-10 in Appendix C.2. We did not observe clear, consistent trends for PReLU⁺ and ReBLU, suggesting that the behavior of activation functions on positive inputs is less important to robustness, but provide the results for these activation functions in Appendix C.3.

Robustness to weak adversary increases when activation functions output positive values on negative inputs. From Figure 3, we observe similar trends across both PReLU and PELU. For these activation functions, α controls the behavior on negative inputs. We observe a significant increase in robustness for models at $\alpha < 0$ compared to $\alpha > 0$, although there a decrease in robustness when $|\alpha|$ becomes large. For both PReLU and PELU, $\alpha = 0$ corresponds to ReLU, $\alpha > 0$ outputs negative values for negative inputs, and $\alpha < 0$ outputs positive values on negative inputs (See Figure 2). This trend suggests that designing a PAF with a parameter that controls the shape of the activation function on negative inputs may be conducive for robustness. Specifically, we will add a parameter

which allows for the PAF to vary the magnitude of positive outputs on negative inputs in a way that is similar to PReLU or PELU (Section 3.3).

Robustness to weak adversary increases when activation functions have high finite curvature.

From Figure 3, we observe similar trends between PSiLU and PSoftplus. For these activation functions α controls the curvature, the maximum value of the second derivative. As α increases, the curvature also increases. First, we note that the models with highest robustness have $\alpha > 1$, where $\alpha = 1$ models their nonparametric variants commonly used in training neural networks. At higher values of α , the shapes of these activation functions grow close to the shape of ReLU which has curvature of infinity. We find that for both PSiLU and PSoftplus, robustness initially increases as α increases and then decreases after a certain point, with this trend being highly significant for PSiLU. This suggests that designing a PAF with a parameter that controls the curvature of the activation function similar to PSiLU and PSoftplus may also benefit robustness.

3.3 PUTTING IT TOGETHER: PSSiLU

We now combine the properties observed to enhance robustness into an activation function. One PAF exhibiting both properties is PSoftplus; however the trend in robustness was not as significant for PSoftplus as for PSiLU, PReLU, or PELU. Since we observed a more significant trend in robustness over parameter α for PSiLU, we introduce a new activation function that is based off of PSiLU. We call this new activation function Parametric Shifted SiLU (PSSiLU) defined as:

$$\text{PSSiLU}_{\alpha,\beta}(x) = x(\sigma(\alpha x) - \beta)/(1 - \beta) \quad (4)$$

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. Recall that the shape of PSSiLU at various values of α and β were shown in Figure 1. 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.

We perform a similar evaluation of robustness across the β parameter of PSSiLU with $\alpha = 1$ in Figure 4. We find that as β increases, there is an increase in robustness, which plateaus at around $\beta = 0.3$. Since higher values of β correspond to giving positive output on negative inputs, this trend matches that of PReLU and PELU. In Appendix C.5, we demonstrate that this trend is consistent across architectures (WRN-28-10 and VGG-16) and datasets (CIFAR-100).

4 INVESTIGATING THE PERFORMANCE OF ADVERSARIALLY TRAINED MODELS USING PARAMETRIC ACTIVATION FUNCTIONS

We now combine PAFs with adversarial training to investigate the impact of incorporating parameters into activation functions on adversarial training. We experiment with the activation functions from Section 3.1 with learnable parameters. 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, ResNet-18, VGG-16 architectures and on CIFAR-10, CIFAR-100, and Imagenette datasets. For CIFAR-10, we also experiment with using additional data during training. For additional CIFAR-10 data, we use DDPM-6M (Nakkiran et al., 2020), a set of 6M CIFAR-10 images generated by DDPM, a generative model (Ho et al., 2020). We note that DDPM-6M does not require any additional real data since DDPM is trained directly on CIFAR-10. DDPM generated samples have been shown to improve the robustness of adversarially trained

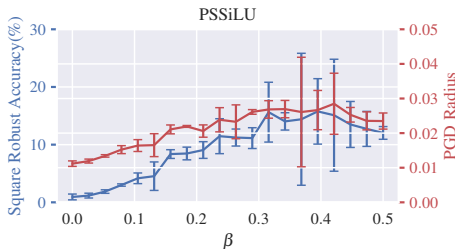


Figure 4: Square robust accuracy and average minimum PGD radius for SSiLU at varied β for ResNet-18 models trained on CIFAR-10. Results are computed over 3 trials.

models (Sehwag et al., 2021; Rebuffi et al., 2021). For the bulk of our experiments, we use 10-step PGD adversarial training (Madry et al., 2018) and focus on ℓ_∞ attacks, but we include results with TRADES adversarial training (Zhang et al., 2019) and ℓ_2 attacks in the Appendix. We include additional details about experimental setup in Appendix A.2.

4.1 THE IMPORTANCE OF ADDITIONAL DATA

We present results for ℓ_∞ attacks on ResNet-18 in Table 2a and ℓ_∞ attacks on WRN-28-10 in Table 2b. We also report results for VGG-16, ResNet-18 models trained with TRADES (Zhang et al., 2019), and ResNet-18 models under an ℓ_2 adversary in Appendix D.

We find that when trained on CIFAR-10 without extra data, most PAFs are unable to outperform their nonparametric variant. For example, on ResNet-18 (Table 2a), PELU has 3.20% lower robust accuracy than ELU when trained on CIFAR-10. Similarly, PSiLU obtains 1.9% lower robust accuracy compared to SiLU on CIFAR-10. This trend can also be seen for WRN-28-10 (Table 2b). Since parametric activation functions are able to take the shape of their nonparametric variants, this suggests that PAFs may need additional regularization to allow the model to converge to a better minimum.

We find that additional data, which can be synthetic as in the case of DDPM-6M, helps regularize PAFs during training, allowing PAFs to outperform their nonparametric variants. For instance, we find that when trained with additional DDPM-6M data, PELU outperforms ELU by 1.70% and PSiLU outperforms SiLU by 1.13% on ResNet-18. A similar trend also holds for WRN-28-10, where PELU outperforms ELU by 8.11% and PSiLU outperforms SiLU by 5.27%.

Activation	CIFAR-10		+DDPM-6M		Activation	CIFAR-10		+DDPM-6M	
	Natural	AA	Natural	AA		Natural	AA	Natural	AA
ReLU	82.29	44.58	82.83	53.67	ReLU	83.39	45.98	85.92	59.27
PReLU	80.16	43.53	83.27	53.66	PReLU	82.75	43.62	86.04	58.74
ELU	81.85	46.76	82.47	51.59	ELU	79.66	45.85	81.09	50.79
PELU	80.37	43.56	83.07	53.29	PELU	83.32	43.85	85.83	58.90
Softplus	80.46	44.64	79.44	49.41	Softplus	79.99	44.41	78.86	49.14
PSoftplus	80.40	44.48	84.56	56.78	PSoftplus	82.94	46.68	86.60	60.94
PReLU ⁺	79.77	42.34	83.63	54.21	PReLU ⁺	81.71	45.05	86.05	59.13
ReBLU	81.19	44.91	83.64	53.74	ReBLU	83.16	46.93	86.39	59.62
SiLU	82.53	46.78	83.53	54.07	SiLU	84.17	47.51	84.90	55.10
PSiLU	80.54	45.45	84.73	55.20	PSiLU	82.41	47.03	86.47	60.37
PSSiLU	81.85	44.70	84.79	58.21	PSSiLU	86.02	48.26	87.02	61.96

(a) ResNet-18

(b) WRN-28-10

Table 2: Natural and robust accuracy of PGD adversarially trained models of various activation functions 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.

To further investigate the impact of additional data on PAFs, we measure the highest train and test robust accuracy using PGD-10 achieved during training. We plot these values in Figure 5 for CIFAR-10 and CIFAR-10+DDPM-6M. PAFs can achieve higher train accuracy compared to nonparametric activation functions, showing that PAFs improve on the expressivity of the model and allow the model to fit to the training set better during adversarial training. However, we also observe that without the additional DDPM-6M data, PAFs are unable to generalize well to the test set, suggesting that the potential of PAFs is locked behind the use of additional training data.

4.2 IMPROVING ROBUST PERFORMANCE THROUGH REGULARIZING PSSiLU

The ability of PAFs to quickly overfit to the data suggests that regularization may improve the performance of PAFs. Unlike the other PAFs, PSSiLU introduces 2 additional parameters into training, which allows it to more easily overfit to the training data compared to other activation functions. Thus, we regularize the value of β in PSSiLU by adding $\lambda|\beta|$ to the loss. We choose $\lambda = 10$ as the default value and show the effect of varying lambda in Appendix D.6. Results for regularized PSSiLU are displayed in all tables as the PSSiLU entry.

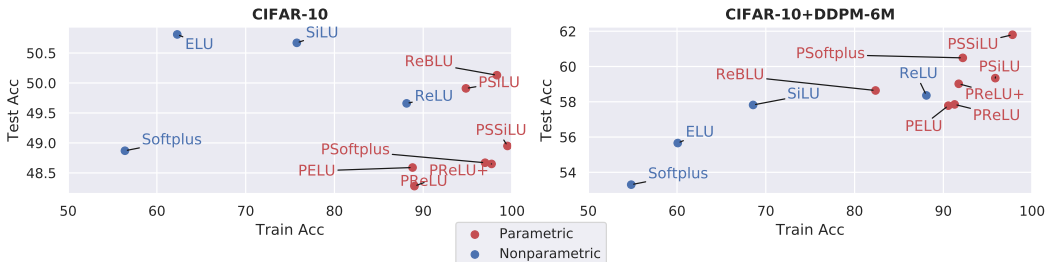


Figure 5: Highest PGD train and test accuracy for ResNet-18 models. Train accuracy is measured only on CIFAR-10.

Achieving state of the art robust accuracy with PSSiLU and DDPM-6M. 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*. This accuracy is only 1.06% lower than the result for WRN-28-10 model with ReLU activations in Table 2b, but WRN-28-10 has 25.3M more parameters than ResNet-18. In other words, we bridged a 5.60% performance gap produced by 25.3M additional parameters by 4.54% by adding only 2 parameters into the 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 (Croce & Hein, 2020). **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 (Gowal et al., 2020).**

Consistency of the PSSiLU family. The function class of PSSiLU captures that of PSiLU and SiLU. PSiLU can be thought of as PSSiLU with $\beta = 0$, which can be achieved by placing a large regularization term on $|\beta|$. Similarly, SiLU is PSSiLU with $\alpha = 1$ and $\beta = 0$, and can be achieved by PSSiLU with a large regularization term on $|\alpha - 1|$ and $|\beta|$. Across datasets and architectures tested, we find that a member of the PSSiLU family is able to consistently obtain high robust accuracy. We also find that this also generalizes to TRADES adversarial training, ℓ_2 attacks, and other sources of additional data (Appendix D). This suggests that the function class of captured by PSSiLU works well in conjunction with adversarial training in improving robustness.

Consistency of smooth PAFs. We find that smooth PAFs (PSoftplus, PSiLU, and PSSiLU) often *improve robust accuracy over ReLU* even when additional data is not present. In Appendix D, we find that this pattern holds across datasets and architectures and generalizes to TRADES adversarial training and ℓ_2 attacks.

4.3 VISUALIZING LEARNED SHAPES OF PARAMETRIC ACTIVATION FUNCTIONS

Previously, in Section 3, we searched for a parameterization of activation function shape that controls factors which **are correlated with** robustness independently of adversarial training. This begs the question, how are the properties observed (positive outputs on negative inputs, high curvature) related to shapes learned through adversarial training? In this section, we visualize the shapes of PAFs learned through the adversarial training objective.

We present the learned shapes of PReLU, PELU, PSiLU and PSoftplus in Figure 6. Additionally, we present the learned shape of PSSiLU in Figure 7.

Relation to Trends from Section 3. We observe that across architectures and datasets, PAFs optimize towards the same qualities that were found to improve robustness for standard trained models: positive behavior on negative inputs and high curvature. In Figure 6, we can see that for all models, PReLU and PELU optimize to give positive output on negative inputs while PSiLU and PSoftplus both optimize towards the shape of ReLU. This suggests that these patterns in shape observed can also help the model better fit the training data.

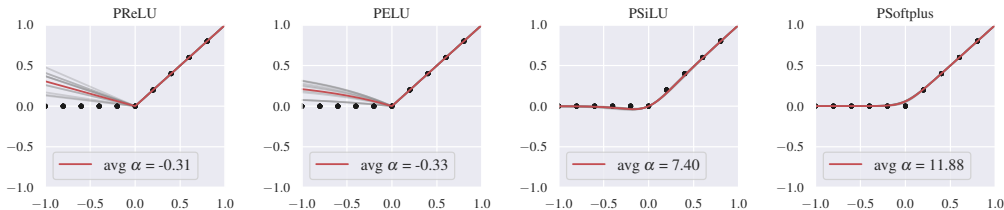


Figure 6: Learned shapes of PAFs across 11 models of various architectures (WRN-28-10, ResNet-18, VGG) trained using PGD adversarial training on various datasets (CIFAR-10, CIFAR-100, ImageNette). Each grey line represents the shape learned by a single model. The red line represents the average of the learned α s across all models. The dotted black line represents ReLU.

Optimization itself improves robust performance. In Figure 7, we visualize the shapes learned by PSSiLU with regularized β . Surprisingly, we find that the β parameters optimize to be 0, which reduces PSSiLU to PSiLU. However, compared to PSiLU, we observed from Table 2 that with additional DDP data, regularized PSSiLU significantly improves robustness over PSiLU. Specifically, PSSiLU improves performance over PSiLU by 3.01% on ResNet-18 and by 1.59% on WRN-28-10. These results suggest that in addition to increasing the size of the function class, adding parameters can also help with optimization, allowing adversarial training to converge to a better minimum.

5 LIMITATIONS AND FUTURE DIRECTIONS

In our work, we showed that by parameterizing activation functions in DNNs, we can improve robust accuracy obtained through adversarial training. **However, we found that PAFs can overfit, motivating the need for regularization.**

Sources of additional data. We find that using additional data boosts the performance of adversarial training with PAFs. Previous works have indicate that current generative models are unable to provide high quality samples for adversarial training on large-scale datasets such as ImageNet (Sehwag et al., 2021). Additional progress on generative models would also improve the performance of adversarial training with PAFs.

Regularizing Parameters. When additional data is not available, regularization may help prevent overfitting. In this work, we use L1 regularization on the β parameter of PSSiLU. A future direction of this work is to explore other forms of regularization such as curvature regularization or Lipschitz constant regularization which are related to robustness (Singla et al., 2021; Moosavi-Dezfooli et al., 2019; Qin et al., 2019; Pauli et al., 2021).

Combining PSSiLU with other training pipelines. Our training pipeline is based off of Sehwag et al. (2021)’s pipeline for PGD training with additional synthetic data. Another future direction of this work is combining PSSiLU with other training pipelines which achieve higher robust performance on RobustBench (Croce et al., 2020) to see if we can further improve robust performance.

6 CONCLUSION

In this work, we studied the impact of parameterizing activation functions on robustness through adversarial training. 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 potential of using learnable PAFs for enhancing robustness of machine learning against adversarial examples.

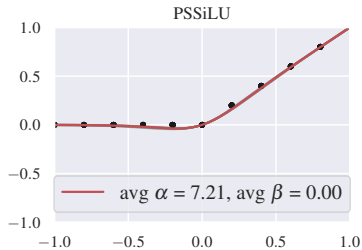


Figure 7: Learned shapes of activation functions across 11 models of various architectures (WRN-28-10, ResNet-18, VGG) trained using PGD adversarial training on various datasets (CIFAR-10, CIFAR-100, ImageNette). Each grey line represents the shape learned by a single model. The red line represents the average of the learned α s and β s across all models. The dotted black line represents ReLU.

7 REPRODUCIBILITY STATEMENT

To ensure reproducibility, we have provided details on experimental setup in Appendix A. Additionally, we have provided our code in the supplementary.

REFERENCES

- 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*, pp. 484–501. Springer, 2020.
- 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*, pp. 274–283. PMLR, 2018.
- 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.
- Sébastien Bubeck and Mark Sellke. A universal law of robustness via isoperimetry. *arXiv preprint arXiv:2105.12806*, 2021.
- Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE symposium on security and privacy (sp)*, pp. 39–57. IEEE, 2017.
- 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*, pp. 11192–11203, 2019.
- 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 (eds.), *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>.
- Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pp. 2206–2216. PMLR, 2020.
- Francesco Croce, Maksym Andriushchenko, Vikash Sehwal, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. *arXiv preprint arXiv:2010.09670*, 2020.
- 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*, pp. 472–478, 2001.
- Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 315–323. JMLR Workshop and Conference Proceedings, 2011.
- Luke B. Godfrey. An evaluation of parametric activation functions for deep learning. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pp. 3006–3011, 2019. doi: 10.1109/SMC.2019.8913972.
- Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun (eds.), *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>.
- 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.

- 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*, pp. 1026–1034, 2015.
- 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*, pp. 770–778, 2016.
- 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 (eds.), *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/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html>.
- Jeremy Howard. Imagewang. URL <https://github.com/fastai/imagenette/>.
- Matt Jordan and Alexandros G Dimakis. Exactly computing the local lipschitz constant of relu networks. *arXiv preprint arXiv:2003.01219*, 2020.
- Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.
- Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with warm restarts. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017. URL <https://openreview.net/forum?id=Skq89Scxx>.
- 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.
- Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Jonathan Uesato, and Pascal Frossard. Robustness via curvature regularization, and vice versa. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9078–9086, 2019.
- 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.
- Tianyu Pang, Xiao Yang, Yinpeng Dong, Hang Su, and Jun Zhu. Bag of tricks for adversarial training. In *International Conference on Learning Representations*, 2020.
- 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.
- Patricia Pauli, Anne Koch, Julian Berberich, Paul Kohler, and Frank Allgower. Training robust neural networks using lipschitz bounds. *IEEE Control Systems Letters*, 2021.
- 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 H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL <https://proceedings.neurips.cc/paper/2019/file/0defd533d51ed0a10c5c9dbf93ee78a5-Paper.pdf>.
- 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.

- 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>.
- 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>.
- Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. In *International Conference on Machine Learning*, pp. 8093–8104. PMLR, 2020.
- Vikash Sehwal, 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.
- Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun (eds.), *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/1409.1556>.
- Vasu Singla, Sahil Singla, David Jacobs, and Soheil Feizi. Low curvature activations reduce overfitting in adversarial training. *arXiv preprint arXiv:2102.07861*, 2021.
- 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 (eds.), *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>.
- Mohammadamin Tavakoli, Forest Agostinelli, and Pierre Baldi. Splash: Learnable activation functions for improving accuracy and adversarial robustness. *arXiv preprint arXiv:2006.08947*, 2020.
- 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.
- 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.
- Boxi Wu, Jinghui Chen, Deng Cai, Xiaofei He, and Quanquan Gu. Does network width really help adversarial robustness? *arXiv preprint arXiv:2010.01279*, 2020a.
- Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. In *NeurIPS*, 2020b.
- Cihang Xie and Alan Yuille. Intriguing properties of adversarial training at scale. In *International Conference on Learning Representations*, 2019.
- Cihang Xie, Mingxing Tan, Boqing Gong, Alan Yuille, and Quoc V Le. Smooth adversarial training. *arXiv preprint arXiv:2006.14536*, 2020.
- Yao-Yuan Yang, Cyrus Rashtchian, Hongyang Zhang, Russ R Salakhutdinov, and Kamalika Chaudhuri. A closer look at accuracy vs. robustness. In *NeurIPS*, 2020.
- Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.
- Valentina Zantedeschi, Maria-Irina Nicolae, and Amrith Rawat. Efficient defenses against adversarial attacks. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pp. 39–49, 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 Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *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*, pp. 7472–7482. PMLR, 2019. URL <http://proceedings.mlr.press/v97/zhang19p.html>.

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*, 2020a.

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*, 2020b.

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>.

A EXPERIMENTAL SETUP DETAILS

A.1 STANDARD TRAINING EXPERIMENTAL SETUP ADDITIONAL DETAILS

Models. We train ResNet-18 (He et al., 2016), WideResNet-28-10 (Zagoruyko & Komodakis, 2016), and VGG-16 (Simonyan & Zisserman, 2015) models. For each model, we replace ReLU activations with activation functions described in Section 3.1. For each activation function, we train multiple models, setting the value of the parameter to those shown in Figure 2.

Datasets. We train ResNet-18, WRN-28-10, and VGG-16 models on CIFAR-10 (Krizhevsky et al., 2009). For ResNet-18 models, we also perform experiments on ImageNette (a 10 class subset of ImageNet) (Howard) and CIFAR-100 (Krizhevsky et al., 2009). For ImageNette, we resize images to 224x224 before passing the images into ResNet-18.

Training Setup We train models using SGD with initial learning rate of 0.1 and use cosine annealing learning rate scheduling (Loshchilov & Hutter, 2017). We train all models for 100 epochs and perform evaluation on the model saved at the epoch which has the highest accuracy on the test set. For ResNet-18 models on CIFAR-10, we run 3 trials. For each trial, all models are seeded to the same seed. For all other models we run a single trial.

Evaluation Setup. For measuring adversarial sensitivity, we consider the adversary constrained to an L-infinity budget. To estimate the smallest perturbation radius for misclassification, we perform a binary search on radius size for 4-step PGD (Madry et al., 2018) with step size of 0.0078. We use 4-step PGD to increase the efficiency of the binary search algorithm. For query-restricted black-box adversary, we use Square attack (Andriushchenko et al., 2020; Croce & Hein, 2020) with budget $\epsilon = 0.031$ and compute the robust accuracy on adversarial examples found within 1000 queries. We measure square attack robust accuracy and PGD radius over images in the test set that are initially classified correctly by the model.

A.2 ADVERSARIAL TRAINING EXPERIMENTAL SETUP

Models. We train ResNet-18 (He et al., 2016), WRN-28-10 (Zagoruyko & Komodakis, 2016), and VGG-16 (Simonyan & Zisserman, 2015) models. For each activation function tested, we replace all ReLU within both models with that activation function. We test parametric activations described in Section 3.1 and allow the parameter α (and β for PSSiLU) to be optimized through training. Additionally, we test nonparametric variants of these activation functions: ReLU, ELU, SiLU, and Softplus. We initialize all parametric activation functions to the shapes of their nonparametric variants (ReLU for PReLU and PBLU, ELU for PELU, SiLU for PSiLU and PSSiLU, and Softplus for PSoftplus). For all parametric activation functions, we share the parameter across activations within the network, so PSSiLU adds 2 parameters to the network overall while all other parametric activation functions add 1 parameter.

Datasets. Overall, we experiment with 3 datasets: CIFAR-10 (Krizhevsky et al., 2009), ImageNette (Howard), and CIFAR-100 (Krizhevsky et al., 2009). Additionally, for CIFAR-10 experiments, we consider the setting of training with and without additional data. We consider 2 sources of additional data for CIFAR-10 models: DDPM-6M (Sehwag et al., 2021) and TI-500K (Carmon et al., 2019). DDPM-6M is a synthetic dataset of 6 million images generated by DDPM, a generative model Ho et al. (2020). Previous works have shown that using samples from DDPM can improve robustness through adversarial training (Sehwag et al., 2021; Rebuffi et al., 2021). TI-500K is a subset of TinyImages which matches the distribution of CIFAR-10 and has also been shown to improve robustness when used in adversarial training (Carmon et al., 2019). We train ResNet-18 models on CIFAR-10, ImageNette, and CIFAR-10+DDPM. For ImageNette, we resize images to 224x224 before passing the images into ResNet-18. We train WRN-28-10 and VGG models on CIFAR-10, CIFAR-10+DDPM, and CIFAR-10+TI-500k. We also train WRN-28-10 models on CIFAR-100.

Training Details. For the bulk of experiments, we use PGD adversarial training (Madry et al., 2018) and train models for 200 epochs. We also train a set of ResNet-18 models on CIFAR-10 and L-infinity adversary with TRADES adversarial training Zhang et al. (2019) with $\beta = 0.6$ (Appendix D.3. For all architectures, we train with 10 step PGD with L-infinity budget of 0.031 and step size of 0.0078. For WRN-28-10 models on CIFAR-10, we also train with 10-step PGD with L-2 budget of 0.5 and step size of 0.075. We train models with SGD with learning rate 0.1 and cosine annealing learning rate scheduling. We seed all models to 12345 to control for differences caused by randomness in initialization. For PSSiLU models, we apply regularization on the magnitude of parameter β to restrict the slope on negative inputs so that it remains small. Specifically, for PSSiLU models, we add a $\lambda|\beta|$ term to the loss function where $\lambda = 10$. Additionally, since a small fluctuation of β leads to a large change in activation function shape, we clip the gradients of the β parameter to have norm 0.01.

Evaluation Details. We evaluate models saved at the epoch which had the highest PGD adversarial accuracy on the test set. We perform our final evaluation of robustness using AutoAttack (Croce & Hein, 2020). During evaluation, we use the same adversarial budget that was used to train the model.

B COMPARISON TO RELATED WORK

Comparison to Goyal et al. (2020) and Xie et al. (2020) Similar to our work, Goyal et al. (2020) and Xie et al. (2020) investigated the impact of varying activation function on the performance of adversarial trained models. Xie et al. (2020) found that smooth activation functions generally outperform nonsmooth activation functions on ImageNet. Meanwhile Goyal et al. (2020) found that SiLU activations perform well and were unable to observe the same trend between activation function smoothness and adversarial robustness. We differ in these works by using learnable PAFs and find that smooth PAFs can consistently outperform ReLU, while their nonparametric variants may not be able to due to low training accuracy.

Comparison to Singla et al. (2021) Singla et al. (2021) explore the impact of curvature on robust generalization. They find that activation functions with lower curvature experience less robust overfitting. In contrast, our work focuses on the impact of activation functions on robustness. We find that activation functions tend towards shapes with higher curvature during adversarial training suggesting that higher curvature allows for better fitting of the training set, which allows for higher robustness on the test set. However, we also observe more overfitting, which agrees with Singla et al. (2021)’s findings, motivating the use of additional data to regularize PAF parameters.

Comparison to Tavakoli et al. (2020) Tavakoli et al. (2020) introduce PAF called SPLASH which can model symmetric piecewise linear activations. They find that the symmetric nature of SPLASH allows standard trained models using SPLASH to be more robust. Unlike Tavakoli et al. (2020), we choose a variety of PAFs with nonlinear components such as PELU, PSiLU, and PSoftplus and combine PAFs with adversarial training. Additionally, we explore the case in which our PAFs minimally add to the set of learnable parameters; while SPLASH introduces as many as 12 parameters at each layer of the network, we introduce 1-2 parameters into the network overall and find that this small increase in parameter number can increase robustness.

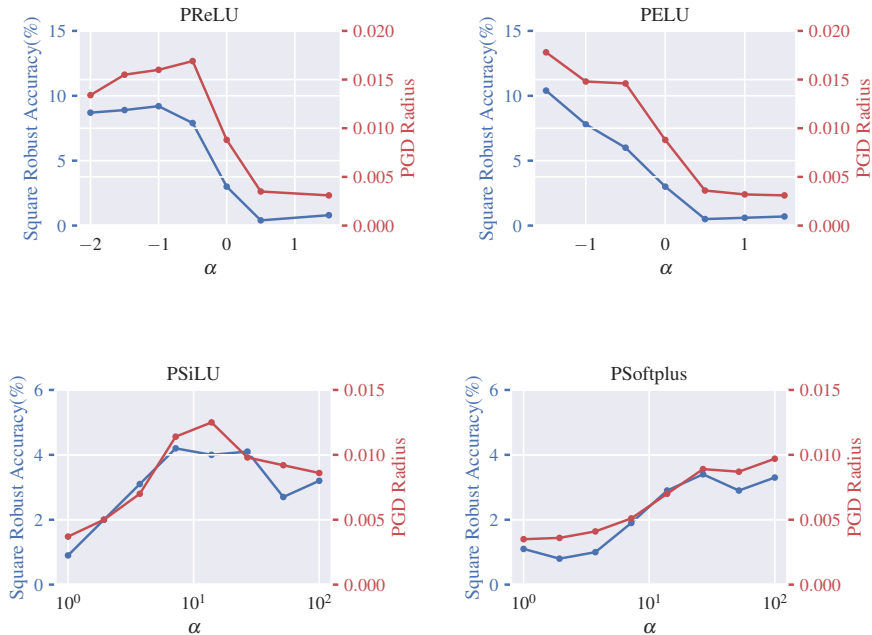


Figure 8: Square robust accuracy and average minimum PGD radius for WRN-28-10 models trained on CIFAR-10 with various parameter α .

C ADDITIONAL RESULTS FOR STANDARD TRAINED MODELS

C.1 CLEAN ACCURACIES OF STANDARD TRAINED MODELS

We report the minimum and maximum classification accuracies for each model and dataset combination across all activations and parameter values in Table 3

Model	Dataset	Min Acc	Max Acc
ResNet-18	CIFAR-10	89.1	95.3
WRN-28-10	CIFAR-10	89.2	96.0
VGG-16	CIFAR-10	76.4	94.1
ResNet-18	Imagenette	79.4	92.7
ResNet-18	CIFAR-100	70.2	77.8

Table 3: Minimum and maximum values for clean accuracy of standard trained models across all activations and parameter values tested.

C.2 GENERALIZATION OF OBSERVED TRENDS

To test whether the trends seen in Section 3 generalize to other model architectures, we repeat experiments on WRN-28-10 (Figure 8) and VGG-16 (Figure 9). We find that across architectures, the trends for PReLU, PELU, PSiLU, and PSoftplus are consistent.

To test whether these patterns also generalize across dataset, we repeat experiments on CIFAR-100 (Figure 10) and ImageNette (Figure 11). We find that these trends are clear for CIFAR-100 but are not clear in ImageNette. For ImageNette, we find that there is little variation across PGD radius and Square robust accuracy is not always correlated with the measured radius as is observed for CIFAR-10 and CIFAR-100.

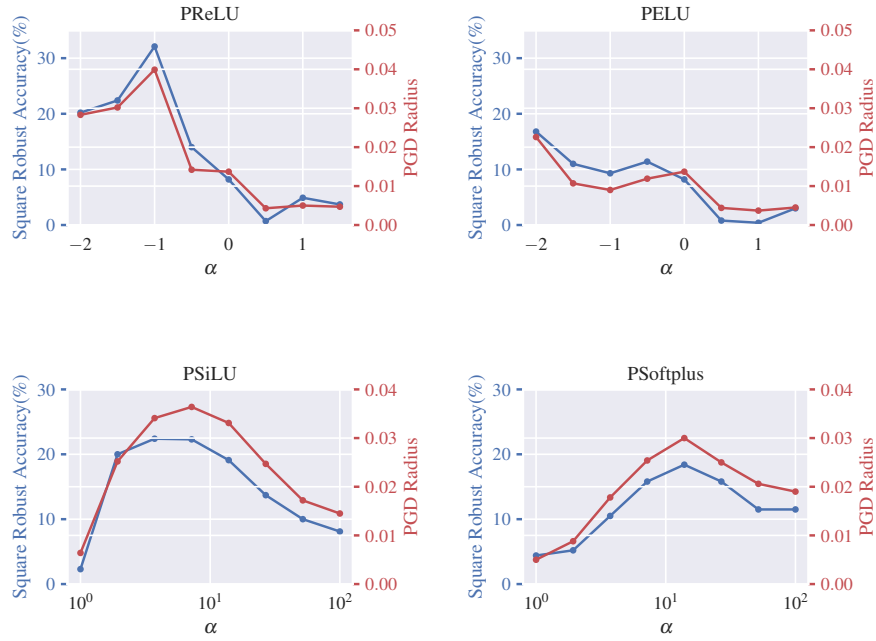


Figure 9: Square robust accuracy and average minimum PGD radius for VGG-16 models trained on CIFAR-10 with various parameter α .

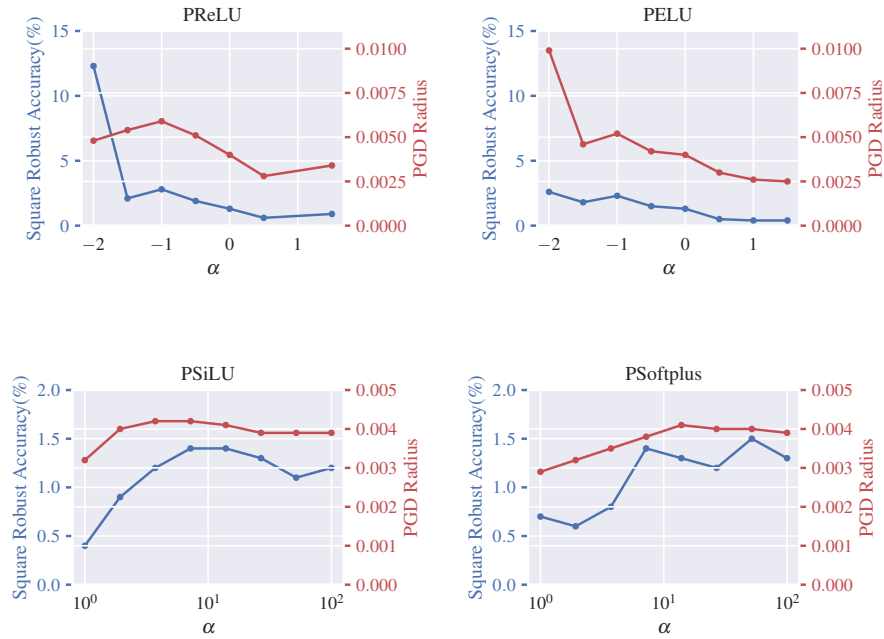


Figure 10: Square robust accuracy and average minimum PGD radius for ResNet-18 models trained on CIFAR-100 with various parameter α .

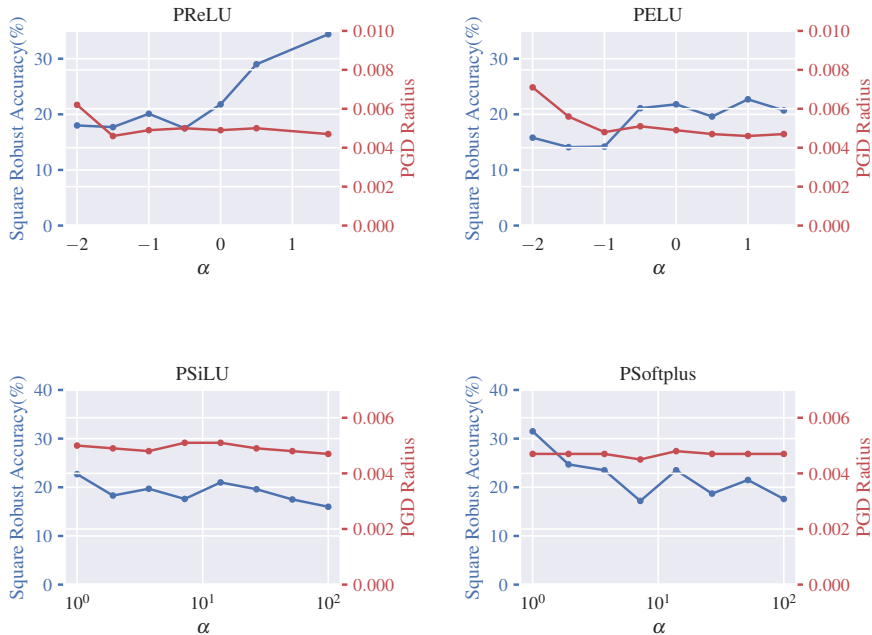


Figure 11: Square robust accuracy and average minimum PGD radius for ResNet-18 models trained on ImageNette with various parameter α .

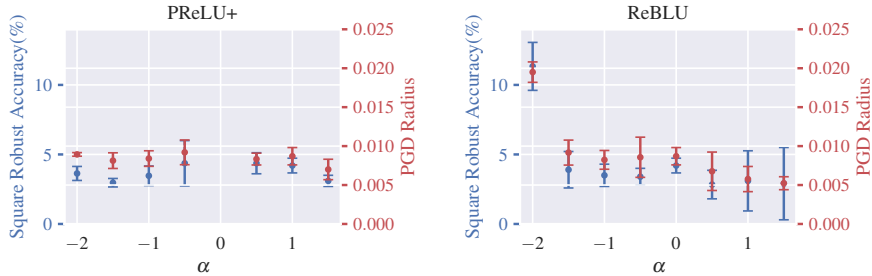


Figure 12: Square robust accuracy and average minimum PGD radius for ResNet-18 models trained on CIFAR-10 with various parameter α for PReLU⁺ and PBLU activations. Errors bars are computed over 3 trials.

C.3 BEHAVIOR FOR ACTIVATION FUNCTIONS VARYING IN THE POSITIVE REGIONS

In addition to testing activations with varied behavior on negative inputs and around 0, we also measure the square robust accuracy and average minimum PGD radius for PReLU⁺ and PBLU. For these activation functions, we were unable to observe any clear trends for the impact of positive behavior. We provide the plot for ResNet-18 models trained on CIFAR-10 in Figure 12. We find that PReLU exhibits high variance in behavior making trends unclear. Figure 12 suggests that $\alpha < 0$ may lead to higher perturbation stability, we find that this is inconsistent across architectures. For instance, we observe the opposite trend in Figure 9 which shows the behavior of PReLU and PBLU for VGG-16 models trained on CIFAR-10. There are no consistent trends across these activation functions, which suggests that positive behavior is less important for robustness. Thus, we do not introduce a parameter to control positive behavior on PSSiLU.

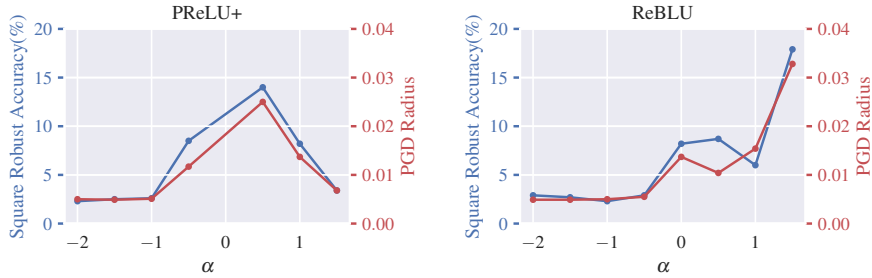


Figure 13: Square robust accuracy and average minimum PGD radius for VGG-16 models trained CIFAR-10 with various parameter α for PReLU⁺ and PBLU activations.

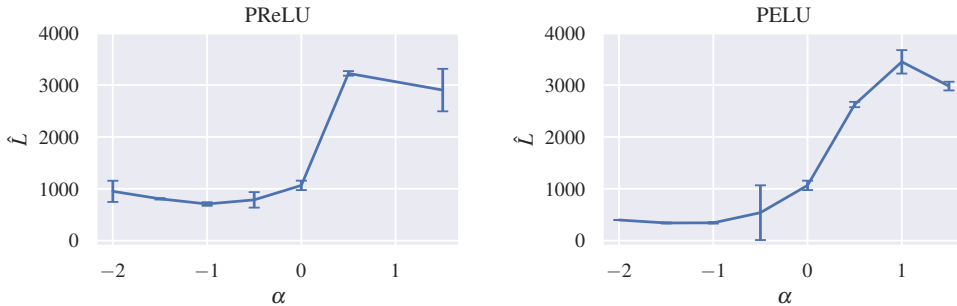


Figure 14: Empirical Lipschitz constant of PReLU and PELU ResNet-18 models trained on CIFAR-10 at varied value of parameter α . Lower empirical Lipschitz constant suggests that the model outputs are more stable in the presence of perturbations. The trend in Empirical Lipschitz constant matches the trends observed in PGD radius and Square robust accuracy.

C.4 EMPIRICAL LIPSCHITZ CONSTANT OF PRELU AND PELU MODELS

For PReLU and PELU, we observe that for $\alpha < 0$ when the magnitude of α becomes large, the adversarial difficulty decreases. We hypothesize that this trend is due to neural network Lipschitz constant. When $|\alpha|$ grows, the Lipschitz constant for PReLU and PELU also increases. The Lipschitz constant of neural network controls the amount of change that can occur in the output when an input is perturbed, so restricting the magnitude of the Lipschitz constant can improve adversarial robustness (Qin et al., 2019; Jordan & Dimakis, 2020; Pauli et al., 2021). Neural network Lipschitz constant depends on the Lipschitz constant of activation functions and weight matrices within the network. We hypothesize that as $|\alpha|$ becomes large, the Lipschitz constant of the neural network increases due to the increase in Lipschitz constant of the activation function. To test this, we measure the empirical Lipschitz constant of PReLU and PELU models, where the empirical Lipschitz constant of a model is defined as (Yang et al., 2020)

$$\hat{L} = \frac{1}{n} \sum_{i=1}^n \max_{\hat{x}_i \in B(\epsilon, x_i)} \frac{\|f(x_i) - f(\hat{x}_i)\|_1}{\|x_i - \hat{x}_i\|_\infty} \tag{5}$$

where f is the model, $\{x_i\}_{i=1}^n$ represent the data inputs and $B(\epsilon, x_i)$ represents a ball of radius ϵ around x_i . \hat{x}_i can be generated by an adversarial attack. We compute this quantity using PGD-10 with radius 0.031 and step size 0.0078 to generate \hat{x}_i . The trends for the empirical Lipschitz constant of PReLU and PELU ResNet-18 models is shown in Figure 14. We find that the trends for empirical Lipschitz constant are also consistent with the trends for Square robust accuracy and PGD radius.

C.5 GENERALIZATION OF TRENDS FOR PSSiLU

To show that the patterns observed on PSSiLU at varied parameter β are consistent across architecture and dataset, we report results for Square robust accuracy and PGD radius on WRN-28-10 and VGG-16 architectures in Figure 15. Additionally, we report results on CIFAR-100 in Figure 16. We find that as β increases the model robustness also increases. This is consistent with our findings on ResNet-18 models trained on CIFAR-10.

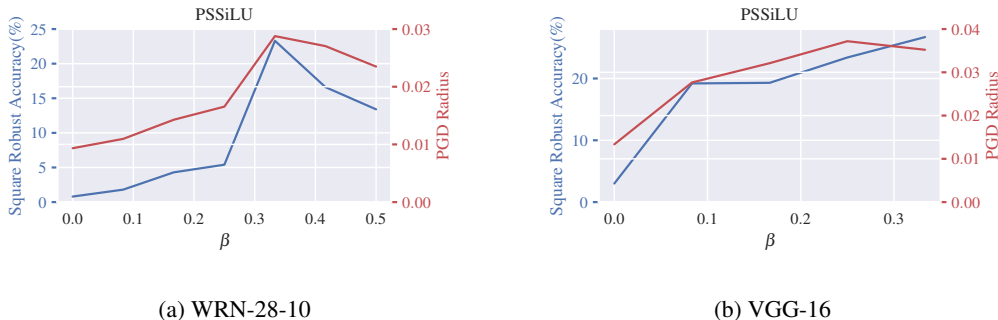


Figure 15: Square robust accuracy and average minimum PGD radius for PSSiLU across parameter β for standard trained WRN-28-10 and VGG-16 architectures.

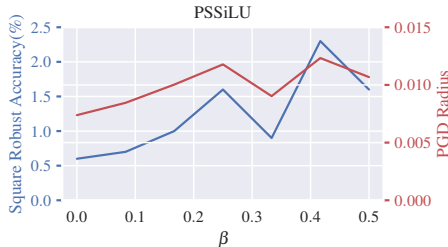


Figure 16: Square robust accuracy and average minimum PGD radius for PSSiLU across parameter β for standard trained ResNet-18 models on CIFAR-100.

D ADDITIONAL RESULTS FOR ADVERSARIALLY TRAINED MODELS

D.1 RESULTS ON VGG-16

We report additional results for VGG-16 models trained on CIFAR-10 in Table 4. We find that trends in VGG-16 are generally consistent with those for ResNet-18 and WRN-28-10: the best performing models are within the PSSiLU family and parametric activation functions outperform nonparametric activation functions when there is additional data in training. Additionally, we find that smooth PAFs (PSoftplus, PSiLU, PSSiLU) improve robust accuracy over ReLU, even without extra data from DDPM-6M.

D.2 RESULTS FOR L2 ADVERSARY

We report results for ResNet-18 models on CIFAR-10 with an L2 adversary in Table 5. We find that the PSiLU achieves the highest robust accuracy both with and without extra data, and PSSiLU has performance comparable to that of PSiLU in both instances. Additionally, we observe that smooth PAFs (PSoftplus, PSiLU, and PSSiLU) all improve robust accuracy over ReLU even without additional data.

Activation	CIFAR-10		+DDPM-6M	
	Natural	AA	Natural	AA
ReLU	76.3	41.5	82.0	51.3
PReLU	78.8	40.3	82.2	51.3
ELU	77.6	40.5	79.9	48.2
PELU	77.4	41.1	81.6	51.2
Softplus	71.9	40.2	75.5	43.8
PSoftplus	81.0	41.9	82.7	53.0
PPRELU	78.5	41.6	82.1	51.6
ReBLU	77.9	41.7	82.3	52.4
SiLU	80.5	43.1	82.3	51.0
PSiLU	77.7	42.5	83.0	53.2
PSSiLU	77.7	41.9	82.8	52.9

Table 4: Natural and robust accuracy of adversarially trained VGG-16 models of various activation functions with respect to ℓ_∞ attacks with radius 0.031 generated through AutoAttack. We highlight robust accuracies higher than ReLU in purple.

Activation	CIFAR-10		+DDPM-6M	
	Natural	AA	Natural	AA
ReLU	89.6	65.1	89.4	74.4
PReLU	84.6	57.5	88.7	71.7
ELU	88.6	65.6	88.0	71.2
PELU	88.7	63.8	89.1	73.9
Softplus	87.3	64.2	86.5	67.7
PSoftplus	89.8	67.1	89.7	75.3
PPRELU	88.5	63.4	88.7	73.9
ReBLU	88.3	65.9	89.2	73.2
SiLU	87.6	64.6	89.4	73.4
PSiLU	89.3	67.7	89.6	75.8
PSSiLU	89.6	67.3	89.9	75.6

Table 5: Natural and robust accuracy of adversarially trained WRN-28-10 models on CIFAR-10 with respect to ℓ_2 attacks with radius 0.5 generated through AutoAttack. We highlight robust accuracies higher than ReLU in purple.

D.3 RESULTS ON TRADES TRAINED MODELS

We report results for ResNet-18 models trained with TRADES adversarial training in Table 6. We find that PSSiLU is able to obtain the highest accuracy on CIFAR-10 without additional data, outperforming ReLU by 0.9%. With additional data, PSiLU obtains the highest robust accuracy (PSSiLU obtains comparable robustness), outperforming ReLU by 1.9%. This is consistent with our observation that the PSSiLU family is able to achieve high robustness. Additionally, we find that PSoftplus, PSiLU, and PSSiLU all improve over ReLU even without additional data during training. This is consistent with our observation that smooth PAFs often outperform ReLU.

D.4 RESULTS ON ADDITIONAL DATASETS

In Table 7, we present the results for ResNet-18 models trained on ImageNette and WRN-28-10 models trained on CIFAR-100. We find that on ImageNette, SiLU achieves the highest robust performance, which is consistent with our finding that members of the PSSiLU family are able to achieve high robust accuracy. We find that PSoftplus achieves the highest robust accuracy on CIFAR-100, but both PSiLU and PSSiLU are able to achieve comparable robust accuracy. Additionally, we find that across both datasets, PSoftplus, PSiLU, and PSSiLU are able to outperform ReLU, further emphasizing our finding that smooth PAFs generally improve over ReLU even without extra data.

We note that we do not use additional data when training for ImageNette and CIFAR-100. The performance of PAFs may further improve if additional data for ImageNette and CIFAR-100 is present.

Activation	CIFAR-10		+DDPM-6M	
	Natural	AA	Natural	AA
ReLU	83.1	48.5	82.2	56.2
PReLU	79.6	45.4	82.8	56.1
ELU	77.5	44.5	77.8	50.8
PELU	82.6	47.0	82.4	55.8
Softplus	73.9	41.4	76.7	47.5
PSoftplus	81.6	49.1	82.6	57.2
PPRELU	83.5	47.7	82.3	55.8
ReBLU	82.0	46.7	81.7	54.9
SiLU	81.1	49.0	79.5	53.7
PSiLU	81.7	49.3	82.8	58.1
PSSiLU	81.2	49.4	82.7	57.9

Table 6: Natural and robust accuracy of TRADES adversarially trained ResNet-18 models of various activation functions with respect to ℓ_∞ attacks with radius 0.031 generated through AutoAttack. We highlight robust accuracies higher than ReLU in purple.

Activation	ImageNette		CIFAR-100	
	Natural	AA	Natural	AA
ReLU	88.6	60.5	59.6	23.5
PReLU	69.7	59.5	57.5	21.6
ELU	87.5	61.2	58.1	24.1
PELU	87.2	57.7	58.6	21.9
Softplus	81.5	54.8	57.1	23.1
PSoftplus	88.9	62.8	60.5	24.8
PReLU ⁺	87.4	59.6	58.6	22.7
ReBLU	87.5	58.3	59.5	24.3
SiLU	88.8	64.1	54.3	22.9
PSiLU	88.7	62.4	59.3	24.2
PSSiLU	87.5	61.2	59.7	24.2

Table 7: Natural and robust accuracy of adversarially trained models of various activation functions with respect to ℓ_∞ attacks with radius 0.031 generated through AutoAttack on ImageNette and CIFAR-100. We highlight robust accuracies higher than ReLU in purple.

D.5 FIXING β ON PSSiLU

Unlike other parametric activation functions tested, PSSiLU has 2 learnable parameters. We experiment with fixing the value of β on PSSiLU so that α is the only learnable parameter. Figure 4 shows the trend for adversarial difficulty over β when α is fixed to 1. We find after about $\beta = 0.3$, we do not see much improvement from increasing the value of β . We set β to 0.3 and trained another set of ResNet-18 models using PSSiLU. The results are shown in Table 8.

Activation	CIFAR-10		+DDPM-6M	
	Natural	AA	Natural	AA
ReLU	82.3	44.6	82.8	53.7
PSSiLU	79.2	42.7	84.5	56.8

Table 8: Natural and robust accuracy of PSSiLU model with β fixed at 0.3 with respect to L-infinity attacks with radius 0.031 generated by AutoAttack.

We find that even with fixed β , we are able to achieve high robust accuracy on CIFAR-10 when combined with DDPM-6M; however, it is not as high as with β as a learnable parameter.

D.6 IMPACT OF REGULARIZATION ON PSSiLU PERFORMANCE

We vary the strength of regularization (λ) on β . We observe that there is a significant jump in robust performance from no regularization $\lambda = 0$ to $\lambda = 0.1$ suggesting that PSSiLU needs regularization

to perform well. This makes sense because in our formulation for PSSiLU, we have the constraint that $\beta < 1$ which allows PSSiLU to maintain a ReLU-like shape. We find that the best performing model is produced when $\lambda = 10$.

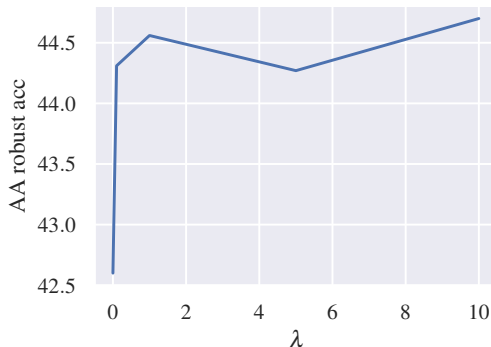


Figure 17: Impact of regularization strength λ on β parameter on AutoAttack robust accuracy of PGD adversarially trained ResNet-18 model.

D.7 LEARNED SHAPES FOR PReLU⁺ AND ReBLU

We present the learned shapes of PReLU⁺ and ReBLU in Figure 18. We find that these activation functions generally optimize so that the slope in the positive region is positive. However, we find that this trend is not consistent across dataset and architecture. For instance in Figure 18, we can see several models which optimize towards negative values of α , leading to negative slope on positive inputs on PReLU⁺ and a downward curve on ReBLU.

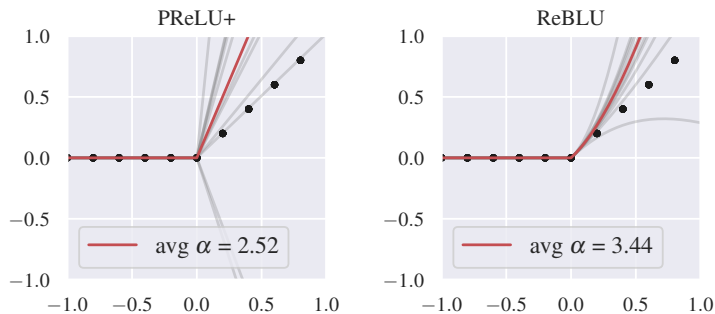


Figure 18: Learned shapes of PReLU⁺ and ReBLU activation functions across all 11 models trained using PGD adversarial training. Each gray line represents the shape learned by a single model. The red line represents the average of the learned α s across all models.