2024 International Conference on Machine Learning and Applications (ICMLA) | 979-8-3503-7488-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICMLA61862.2024.00291

Unifying Robust Activation Functions for Reduced Adversarial Vulnerability with the Parametric Generalized Gamma Function

Sheila Alemany*, Emma Worthington†, Alberto Dominguez‡, Ilan Grapel‡, Niki Pissinou§
*§School of Computing and Information Sciences, Florida International University
†College of Engineering and Applied Science, University of Colorado Boulder
‡National Science Foundation Research Experience for Teachers Fellow
{*salem010, §pissinou}@fiu.edu

***Abstract*—Adversaries minimally perturb deep learning input data to reduce a learning model's ability to produce domain-specific data-driven recommendations to solve specialized tasks. This vulnerability to adversarial perturbations has been argued to stem from a learning model's non-local generalization over complex input data. Given the incomplete information in a complex dataset, a learning model captures non-linear patterns between data points with volatility in the loss surface and exploitable areas of low-confidence knowledge. It is the responsibility of activation functions to capture the non-linearity in data and, thus, has inspired disjointed research efforts to create robust activation functions. This work unifies the properties of activation functions that contribute to robust generalization with the generalized gamma distribution function. We show that combining the disjointed characteristics presented in the literature with our parametric generalized gamma activation function provides more effective robustness than the individual characteristics alone[1].**

I. Introduction

The fast-paced development of machine learning models with exceptional accuracy and generalization has begun an equally intense pursuit of creating robust and resilient systems. One of the critical challenges in this pursuit is the vulnerability of machine learning models, especially deep neural networks, to adversarial attacks [1]. Attackers generate malicious inputs by optimizing their attacks to find a minimal perturbation to an existing input sample based on the model's probability density function and decision surface such that it causes incorrect model output. These adversarial perturbations have been shown to impact real-world domains, especially in safety-critical applications like autonomous vehicles [2], medical diagnoses [3], and mobile/Internet of Things (IoT) systems [4].

In attempts to maintain high-performing accuracy despite these stealthy malicious inputs (e.g., increasing adversarial robustness), existing literature has proposed adversarial training [5], regularization [6], and varying data augmentations [7]. Of existing efforts in realm of adversarial machine learning, adversarial training has been shown to maintain the highest-performing machine learning models [5]. However, it has been found to have poor generalization and a significant increase in training time, keeping the pursuit of adversarial robustness a significant research priority [8]. A common thread among proposed improvements is the importance of the parameter loss and decision surfaces that facilitate or inhibit an adversary's ability to attack effectively.

Activation functions play a significant role in a learning model's created decision surfaces, even in scenarios with comparable high generalization performance. As a result, studies have explored curvature in activation functions to improve the generalization quality of adversarial training and the overall adversarial robustness [9]. Additionally, varying activation functions have been proposed to increase the overall adversarial robustness through different means (e.g., non-monotonicity and symmetry) [10], [11], [12]. Unfortunately, it seems as though these efforts were disjointly executed. Thus, they reached separate, parallel conclusions. As a result, we propose a parametric activation function that unifies the properties of existing robust activation functions and evaluate how consolidating these characteristics impact the overall decision surface from an adversarial robustness perspective.

II. Related Work

a) Loss and Decision Surfaces for Robustness: The geometric representations of knowledge (i.e., input/parameter loss surfaces, decision surfaces) have been studied in machine learning for improved explainability [13], [14]. Researchers have begun similarly analyzing learning models from an adversarially robust perspective [15]. However, similar to the noteworthy research area of explainable AI, there remain substantial questions regarding all the geometric qualities that prevent an adversary's ability to optimize the most uncertain or humanely-unpredictable outputs. Adversaries have efficiently identified these uncertain spaces by traversing the loss and decision surfaces while optimizing their attacks [16], [17]. Thus, successful defenses, such as adversarial training, increase robustness by directly focusing on the uncertainty of the spaces that specific attacks target [18]. Meanwhile, others, such as robust feature selection or dimensionality reduction techniques, aim to minimize the overall model uncertainty

[1] The source code for this research effort: https://github.com/sheilaalemany/generalized-gamma-activation.git

1946-0759/24/$31.00 ©2024 IEEE
DOI 10.1109/ICMLA61862.2024.00291

[19]. These defenses have been shown to increase robustness, but work still needs to be done for an overall better understanding of what we, as a research community, strive for regarding a robust decision surface.

Various efforts have identified smoothness as a favorable characteristic of adversarial robustness. Smoothness refers to the property of a model's loss or decision surface to change gradually and predictably as the input data changes (e.g., higher quality generalization) [20]. Visually, as expected, the surfaces look smooth, with no visual steps. From an adversarial perspective, the lack of smoothness in a decision boundary results in abrupt or erratic changes in its output, resulting in an adversary's ability to quickly identify minimal perturbations that cause the most erratic change to a model's output. On the other hand, smoothness also impacts the transferability of adversarial examples or malicious inputs' ability to fool unseen models with a significant success rate. Trained models with local non-smooth loss surfaces harm the transferability of generated adversarial examples [21]. Overall, smoothness has been achieved by constraining the model's architecture, training process, or loss functions [22]. Defense techniques, such as adversarial training and defensive distillation, have combined the idea of smoothness with other strategies to create models that are more resistant to adversarial attacks [23], [24], [25].

Additionally, a flat loss surface, or the property of a learning model's loss function where the loss function value changes only minimally as its parameters vary, has been positively correlated with robustness [20], [26]. Kanai et al. [20] showed how the flatness of the loss in the input space can contribute to increased smoothness in the decision surface. In adversarial training, a flat loss surface can contribute to stability during the training process, reducing sharp and unpredictable local minimum [26]. This works as a regularization technique that discourages the trained model from fitting to noise or minor, highly uncertain variations in data. Hence, it increases robustness against these imperceptible malicious examples. Yu et al. [15] argued that although a flat loss surface in the input space is valuable under adversarial settings, observing the decision surface contributes more insight into the adversarial robustness of a model since the characteristics in the decision surface correspond more highly to robustness compared to the loss surface. In this work, we observed similar patterns.

b) Robust Activation Functions: Tavakoli et al. [10] proposed SPLASH, a piecewise dynamic linear function optimized for robust generalization during training. Their optimized activation function is non-monotonic and aligns with the results by Zhao et al. [27], which proved the importance of symmetric activations to suppress signals of exceptional magnitude (i.e., more significant perturbations). Meanwhile, Rozsa et al. [11] introduced tent activation functions with bounded open space risk as they observed that adversaries exploit the unbounded open space risk that standard monotonic activation functions provide. Interestingly, they also show that open space risk cascades over each layer to create an overall vulnerable classifier implying that a robust activation must be present at each layer. Although SPLASH was a dynamic function that was different and robustly specific to each layer, they did not elaborate on which activation function properties should be prioritized in early vs. later layers. The main limitation with the approaches by the SPLASH [10] and tent [11] activations is the significant (up to 2x) increase in training time. Parisi et al. [28] reached a similar activation shape as SPLASH without iterative learning that also improved general robustness implying that the non-monotonic nature of the activation function contributed more towards adversarial robustness than the dynamic, unique activations per layer. Lastly, Singla et al. [29] suggested using smoothness and low curvature in activation functions to increase robust generalization, specifically when using adversarial training to avoid overfitting to adversarial examples. Low curvature is defined as relatively small second-derivative values for the activation functions. Dai et al. [12] observed similar robustness effects through low curvature with their ReBLU function. Still, they did not explain its impacts on the decision surface and, thus, the impact of low curvature on adversaries' ability to optimize stealthy malicious inputs. With this work, we aim to unify the research efforts of robust activation functions and take a deeper look into how this changes the decision surface to better understand favorable or unfavorable characteristics from a robustness perspective.

c) Stackable Defenses Against Adversarial Examples: Different from ensemble approaches, stackable defenses are implemented through different sections of the machine learning pipeline. These defenses can be entirely independent but target specific vulnerable components. For instance, detection techniques can be used to filter out malicious inputs with larger perturbation budgets along with dimensionality reduction to reduce uncertainty contributed by non-robust features to defend against malicious inputs with smaller perturbation budgets [30]. In addition to these defenses, adversarial training can be stacked to increase robustness against adversarial examples generated by a specific adversarial attack.

III. Parametric Generalized Gamma Activation Function

Based on our literature review, we have pinpointed the following activation function characteristics as the main favorable ones for increased adversarial robustness: (1) continuous and differentiable in the range $[0, 1]$ and domain $(-\infty, \infty)$ to avoid gradient masking [31]; (2) non-monotonic and symmetric [27], [10]; (3) bounded with finite support (i.e., goes to 0 beyond a certain distance from the origin) [11]; (4) smoothness [29]; and (5) relatively low curvature [29]. We chose to employ the generalized gamma distribution function as an activation function since we identified this as a highly parametric function that allows us to mold an activation function into one that meets our required characteristics thanks to its shape and scale parameters.

Figure 1 shows our generalized gamma activation (GenGamma) function compared to ReLU, Swish, hyperbolic tangent, and robust tent activation functions. It is worth

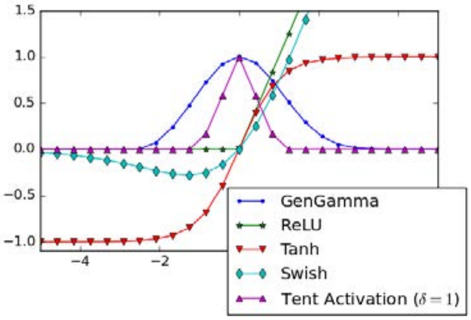


Fig. 1: Comparing the generalized gamma activation function with the ReLU, Swish, hyperbolic tangent, and robust tent activation functions

mentioning that the activation function shape we reached with our parametric generalized gamma function is similar to that of the Gaussian distribution function. Sibi et al. [32] employed the Gaussian activation function as activation functions for optimizing the training process in deep learning. However, the main characteristics of activation functions favorable for increased adversarial robustness remain an open research problem. Thus, we anticipate that our generalized gamma activation function is more adept for this application due to the flexible, parametric nature of its shape and scale parameters as research identifies further desired activation function qualities.

a) Initialization: The generalized gamma distribution has two shape parameters (α, c), and a scale parameter (s). For the activation function to be continuous and differentiable from $(-\infty, \infty)$, we define the generalized gamma activation function as:

$$f(x, \alpha, c) = \frac{|c| x^{c\alpha - 1} e^{-x^c}}{s\Gamma(\alpha)} \tag{1}$$

where $x = \frac{x - \mu}{\beta} \geq 0$, $\alpha > 0$, $c \neq 0$ and $\Gamma(\alpha)$ is the Gamma function on α. For $x < 0$, $f(x, \alpha, c) = 0$. We achieved a range within $[0, 1]$ and function shape that met the implications from past robust activation efforts with the parameters $\alpha = 1$, $c = 3$, $s = 1.17$, $\beta = 3$, and $\mu = -2.6$. Similarly to the tent activations, initialization of the generalized gamma activation function needs to ensure that significant inputs do not fall into saturated regions to avoid low model performance [11]. Observing the convergence of the loss values and of high-performing accuracy, we were able to ensure the deep learning model could still perform comparably compared to the other activations. Given our initialized parameters, we observed comparable accuracy when training on the MNIST and CIFAR-10 datasets against the other activation functions (within 5%). The scale parameter can be increased for more complex models to maintain high-performing accuracy.

IV. Evaluation

A. Activation Functions

a) ReLU: The Rectified Linear Unit (ReLU) activation function is widely used in deep learning models because its simple yet effective definition introduces non-linearity to the network without being computationally exhaustive. This aids the learning model to accurately represent complex relationships found in data with less training time compared to other activation functions [33]. It is defined as:

$$f(x) = max(0, x) \tag{2}$$

The ReLU activation mitigates the vanishing gradient problem, or when gradients become small to the point where they hinder convergence to high-performance accuracy, that can be present when using sigmoid or hyperbolic tangent [34]. Lastly, sparsity is encouraged with this activation because negative inputs are transformed to 0, with only a subset of neurons being activated at any given time. Encouraging sparsity is favorable from an adversarial defense perspective since sparse networks tend to have a lower sensitivity to small changes in input due to their nature of ignoring a significant portion of their input space [35]. This can make it more difficult for adversarial perturbations to impact the network's decision boundary significantly.

b) Swish: The swish activation function was introduced as an alternative to the traditional ReLU [36]. It is defined as:

$$f(x) = x * sigmoid(\beta x) \tag{3}$$

where β is a learnable parameter that determines the slope of the function. The core idea behind the swish function is that it maintains the desirable properties of ReLU (e.g., a positive slope for positive inputs) but introduces a smoothness akin to hyperbolic tangent for negative inputs. In addition to smoothness, swish also contains lower curvature than the existing hyperbolic tangent and sigmoid activations [29].

c) Hyperbolic Tangent: The hyperbolic tangent activation function, often abbreviated as "tanh," is a non-linear function that transforms its input into a range between -1 and 1. It is defined as:

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{4}$$

This activation function is symmetric and centered around zero, which promotes a more balanced representation of data and contributes to a probability density function that is more humanely-predictable across the entire input space [27].

d) Robust Tent: The tent activation was derived based on the core hypothesis that adversaries leverage open space risk of activation functions when generating adversarial examples since when the input strays from the training samples, there is an increased risk that it is from an unknown, highly-uncertain distribution. It is defined as:

$$f(x, \delta) = max(0, \delta - |x|) \tag{5}$$

where δ is the parameter that determines the width of the tent, making the function sensitive to initialization, similar to our parametric generalized gamma activation function. In addition to significantly increasing training time, the tent activation primarily lacks smoothness.

e) SPLASH: The Simple Piecewise Linear and Adaptive with Symmetric Hinges (SPLASH) activation was derived as a parameterized piecewise linear activation function that could approximate a wide range of functions while restricting the function to be continuous and grounded with symmetric and fixed hinges. The number of hinges determines the number of parameters that must be tuned during training. They incorporated the activation into the hidden unit h as:

$$h(x) = \sum_{s=1}^{(S+1)/2} a_+^s max(0, x - b^s) + \sum_{s=1}^{(S+1)/2} a_-^s max(0, -x - b^s) \tag{6}$$

where S is the number of hinges in the activation function. Given this definition per hidden layer, each layer derives a unique function during training. This significantly increases training time (up to 2x) but maintains high benign performance accuracy while also increasing robustness. As stated in the Section II, an activation of similar shape reached comparable robustness as SPLASH hinting that the non-monotonicity ultimately contributed more towards adversarial robustness than the dynamic, unique activations per layer [28].

B. Training and Performance

Our training configuration for this work was LeNet-5, the MNIST [37] and CIFAR-10 [38] datasets, with our activation function requiring approximately 10% more epochs to reach comparable performance (within 5% benign accuracy) to the ReLU, hyperbolic tangent, and swish activation functions on an M1 Macbook Pro. Both datasets were divided into a 75/15 training and testing split. Each activation function under evaluation was included in each hidden layer since robustness has shown to have a cascading effect in the hidden layers [11]. The learning rate was 0.01 for all tests. We selected this training environment to ensure the highest performance across all the models that were trained without modifying any training parameters outside of our control variable of the activation functions. This is because even modifications to the learning rate have been found to influence the optimized loss and decision surfaces [39]. Thus, fine-tuning additional parameters could impact an adversary's ability to attack effectively and impacting the conclusions of this work. Lastly, we could not execute the tent and SPLASH activation functions with comparable performances as reported in the original publications. Still, we compared our results against the documented performances by SPLASH since they had the same evaluation environment with LeNet-5, MNIST, and CIFAR-10.

C. Threat and Attack Methods

To ensure analysis for worst-case scenario robustness, we tested the activation functions using white-box evasion attacks where the adversary has full access to the trained neural network, the defenses used, and the data distribution after training [40]. We consider evasion attacks where the adversaries can attack only during model deployment, meaning they tamper with the input data after the deep learning model is trained.

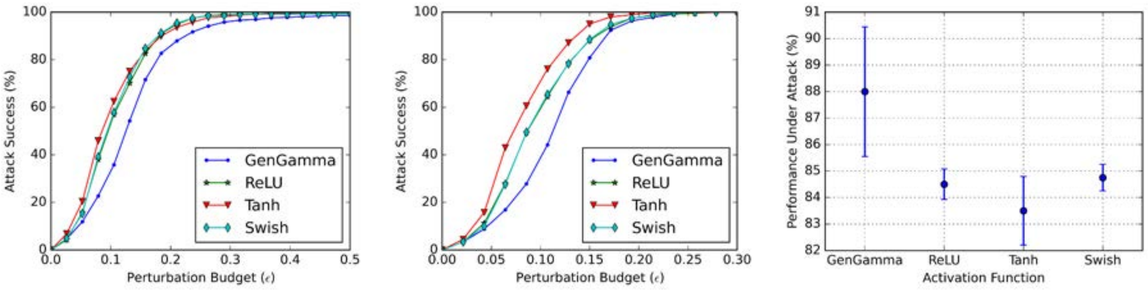
To generate the adversarial examples, we used Fast Gradient Sign Method (FGSM) [41], Projected Gradient Descent (PGD) [42], and C&W l_2 [43] attack implementations from the Adversarial Robustness Toolbox by IBM Research [44] with no changed hyperparameters. These attacks vary in how they traverse the loss or decision surfaces to identify the most stealthy adversarial example possible. FGSM is an efficient, one-step attack that generates adversarial examples using the gradient of the loss function. PGD is an iterative improvement over FGSM that refines perturbations using multiple iterations and projection. C&W results in the stealthiest malicious inputs as it minimizes an objective function that consists of two components: a loss term that encourages misclassification and a term that encourages the perturbed image to be visually similar to the original image.

V. Results

We are focused on the robustness of neural network classifiers, our primary metric is the networks' performance accuracy as a function of the perturbation budget, or distortion (ϵ). Specifically, we define attack success as an adversary's ability to reduce model accuracy: attack success $= 1 -$model accuracy. To ensure that we evaluate an adversary's impact on the accuracy, we only perturb 1,000 random benign input samples that were correctly classified before the model was attacked. Figure 2 compares the activation functions, our generalized gamma activation against ReLU, swish, and hyperbolic tangent, with the MNIST dataset on the LeNet-5 architecture. We observe a significant increase in robustness across a varying range of perturbation budgets for the three gradient-based attacks we employ. For instance, when the perturbation budget is 0.1 for the FGSM attack, our GenGamm activation maintains an attack success of 23% while the other activation functions have an average attack success of approximately 51%. We observe similar significant improvements with the PGD and C&W attacks as shown in Figures 2b and 2c. Figure 2c contains the average distribution of the performance under attack across 15 iterations. Performance under attack refers to the performance accuracy of the LeNet-5 model, given the input is maliciously perturbed. We can see that the consistency of attack performance is more volatile for our generalized gamma activation when compared to the other activations. However, regardless of the less predictable attack results, we remain with higher robustness.

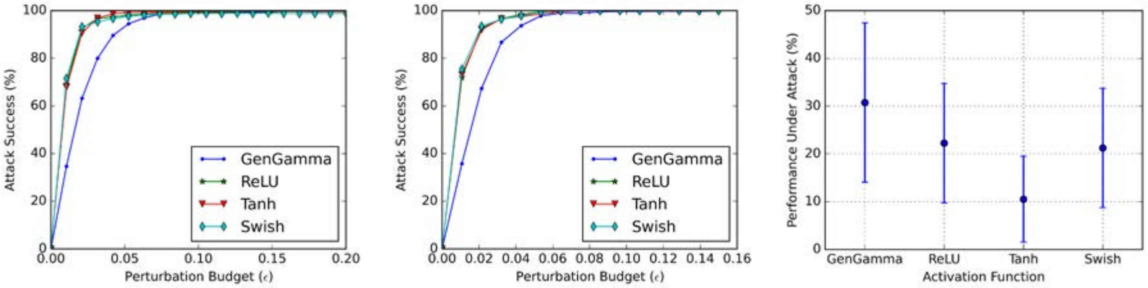
We observe similar patterns when evaluating with the CIFAR-10 dataset. Figure 3 shows that the attacks required less perturbation to achieve high attack success due to the increased model uncertainty with the more complex dataset. However, despite the overall higher attack success rate, we perform significantly better than the other activations, meaning we have a higher robustness rate when there is more model uncertainty. We also observe a 26.3% increase in robustness compared to SPLASH with the CIFAR-10 dataset and a perturbation budget of $e = 0.06$.

A con that is introduced with the generalized gamma activation is an increase in training time. Although the training time was not increased 2-3x times like the SPLASH [10] and tent [11] activation efforts, our training increased by 25%



(a) FGSM (b) PGD (c) C&W l_2

Fig. 2: Comparing performance of different attacks with the MNIST dataset on the LeNet-5 architecture



(a) FGSM (b) PGD (c) C&W l_2

Fig. 3: Comparing performance of different attacks with the CIFAR-10 dataset on the LeNet-5 architecture

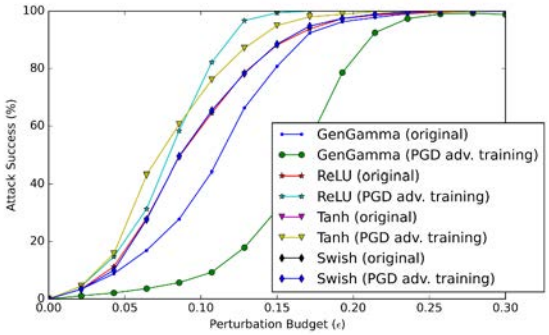


Fig. 4: Comparing performance activation functions under the PGD attacks on the MNIST dataset

relative to the ReLU, hyperbolic tangent, and swish activation functions. However, the parametric nature of the generalized gamma activation function allows us to tailor the shape and scale parameters to improve convergence during training in future work.

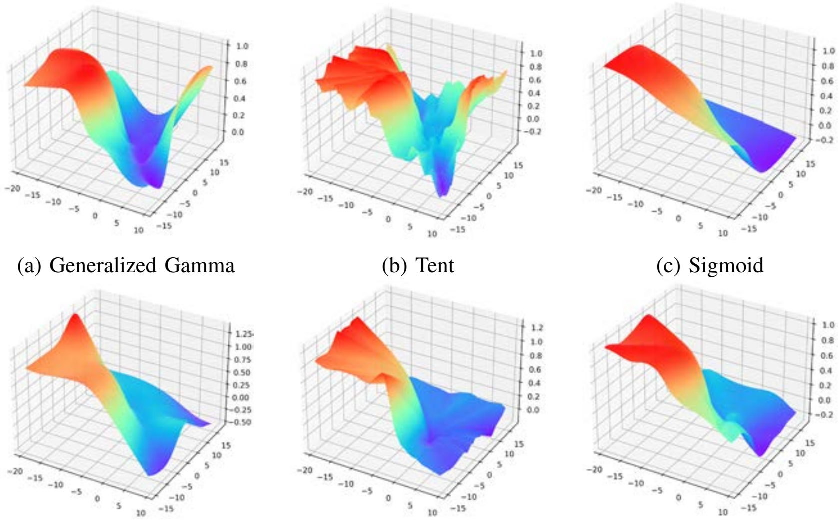
A. *Stacking with Adversarial Training*

We have stacked the activation functions with adversarial training to explore how stacking robust activation functions can complement the existing high robust performance of adversarial training. Figure 4 above compares how the models with and without adversarial training compare with the varying activation functions. We can see how adversarial training with our generalized gamma activation function can increase overall robustness by an average of approximately 33% across all perturbation budgets $\epsilon \in (0, 0.3]$. It is also interesting to note that the generalized gamma activation function still performed better than the other activation functions with adversarial training, highlighting the tremendous impact that a robust activation function can have on controlling the model uncertainty that can be exploited. Swish stacked with adversarial training follows with robust performance after our activation as consistent with the discussion presented by Kanai et al. [24] and Singla et al. [29]; highlighting the impact of smoothness and low curvature on robust generalization of the adversarial examples used during training.

VI. Discussion: Looking at Decision Surfaces

Defining the decision boundary of a neural network from a loss perspective allows us to observe how the trained decision surface is influenced by the activation functions and how adversaries can consequently exploit them with a perturbation budget of $\epsilon \in P$ where P is the set of all possible perturbations. As derived by Yu et al. [15], we can analyze the geometry as follows:

$$L(\theta, x + \Delta x) = L(\theta, x) + J\Delta x + \frac{1}{2!}\Delta x^T H \Delta x \quad (7)$$



(a) Generalized Gamma (b) Tent (c) Sigmoid

(d) Swish (e) ReLU (f) Tanh

Fig. 5: Decision surfaces for six (6) activation functions on a Gaussian mixture data model and binary classification with a 2-layer fully connected neural network

with the following constraint:

$$|L(\theta, x+\epsilon) - L(\theta, x)| < t, \forall \epsilon \in P \tag{8}$$

where x is an input sample with a label y_t corresponding to the correct class t, $L(x)$ is the decision boundary, θ is the parameters of the neural network, J is the Jacobian vector of the same dimension as x and H is the Hessian matrix or the square matrix of second-order partial derivatives of the loss function with regard to x. Combining these equations, the following inequality is derived to observe how the Jacobian vector and Hessian matrix relates to the adversarial perturbation magnitude:

$$max(|J \cdot \epsilon + \frac{1}{2}\epsilon^T H \epsilon|) < t \tag{9}$$

This inequality enforces that the neighborhood points of x should not only share the same decision but also have similar confidence bounded by the absolute, non-negative difference of $L(\theta, x) = t$. Thus, the upper bound of this function defines the exclusive lower bound distance of the adversarial perturbation needed to cause a misclassification. We can see then how the model robustness and the stealthiness of the generated adversarial examples rely heavily on the magnitude of the Jacobian and the eigenvalues of the Hessian.

Visually, we evaluated the decision surface characteristics for the varying activation functions on a Gaussian mixture data model and binary classification with a 2-layer fully connected neural network shown in Figure 5. The motivation behind our discussion is to provide an intuition for our observations on the models' Jacobian or Hessian matrix from an adversarial perspective and not to rigorously prove a monotonic relationship between these activation characteristics and robustness.

a) Symmetric, non-monotonic, and bounded with finite support: The generated abstractions between hidden layers are cascaded across each layer to learn increasingly abstract features, resulting in adversarial vulnerability to cascade across layers [27], [11]. Zhao et al. [27] found that empirical distributions of the abstracted features are often compact and symmetric, causing symmetry to represent the input data distribution better and suppress signals that stray from the expected distribution. Thus, as expected, the impact of symmetry on adversarial robustness depends on the nature of the data and the specific symmetry involved. Some symmetries are naturally present in data, while others may need to be explicitly enforced during training, as done by Wang et al. [45] for instance. The non-monotonic property is required for an activation function to be symmetric.

Activation functions bounded with finite support have been contributed to have the same benefits as symmetric ones in suppressing signals that stray from a limited range [11]. In contrast, non-symmetric and unbounded activation functions like ReLU have no upper bound, which allows for a broader range of values. This change to the Jacobian results in a restricted optimization of adversarial examples as adversaries exploit the steepest change in eigenvectors. These steeper changes are often found in signals that stray from the distribution since those are the areas correlated to high model uncertainty. We are minimizing the magnitude of the Jacobian through our activation function, resulting in less sensitivity to minor changes and increased robustness.

Overall, symmetry and boundedness with finite support can be seen as a form of regularization in neural networks by suppressing signals that stray from the benign distribution. This regularization can help improve generalization and adversarial robustness by encouraging the network to rely on more stable and meaningful features, a characteristic that also benefits adversarial training [46]. We observe both symmetry and boundedness in the tent activation function and our generalized gamma activation function. Figure 5 shows how the symmetry across the y-axis is reflected in the equally symmetric Gaussian mixture data model. However, steep visual steps are observed for the tent activation function in Figure 5b due to its lack of smoothness which negatively impacts robustness.

b) Smoothness and low curvature: Bounded activation functions do not always result in smoother decision boundaries, as we observed with the tent activation function. Activation functions that contain smoothness themselves result in inherently smoother decision surfaces without the need for additional regularization techniques or changes to the loss function. Overall, the outputs of these activation functions change gradually as the input varies, leading to less volatile values in the Jacobian, smoother transitions between classes in the decision surface, and increased robustness as described in Section II.

Singla et al. [29] highlight how smoothness tends towards higher robustness only when the activation function has low curvature since low curvature directly impacts the norm of the Hessian matrix. Increasing the curvature in an activation function increases the maximum eigenvalue of the Hessian matrix and, consequently, decreases the minimal perturbation needed by an adversary to cause a misclassification. We calculated a maximum curvature value of 1.44 for our generalized gamma activation function. Similarly, the maximum curvature for the swish and hyperbolic tangent activation functions were 1.27 and 1.91, respectively. As a result, with the parameters evaluated in this work, the curvature of our function was not lower than swish. The low curvature values and the open space risk (e.g., the metric measure the boundedness with finite support [11]) are inversely proportional, complementing each other concerning adversarial robustness. Future deployments of the generalized gamma activation can observe a reduced curvature to accommodate for the increase in model complexity at the cost of increasing the tight boundedness of the function.

c) Sparsity and non-linearity: In addition to the main characteristics contributing to adversarial robustness, our generalized gamma activation function encourages sparsity and increased non-linearity. Like the ReLU function that was identified as encouraging sparsity, the generalized gamma activation function introduces sparsity by suppressing values that stray from the distribution. This means only a subset of neurons in a layer will activate for a given input, leading to sparse activations. Overall, we are minimizing the magnitude of the Jacobian through our activation function and reducing the trained model's sensitivity to minimal perturbations by encouraging sparsity.

The generalized gamma activation function also has increased non-linearity compared to the other activation functions. Changing the model's non-linearity has alluded to enhance adversarial robustness because a more non-linear decision surface that more closely aligns with the data manifold's non-linearity reduces the learning model's uncertainty [47]. In other cases, non-linearity has been contributed to reducing robustness [48]. We are not contributing "increased non-linearity" to adversarial robustness as it is not a one-size-fits-all solution because the data manifold must be thoroughly understood for each unique dataset to achieve the correct amount of non-linearity. Understanding the linearity of the data manifold in highly complex real-world data is challenging and a highly researched area in generalization in machine learning [49], [50]. However, we mention the change in non-linearity here to be thorough with the potential characteristics that could have influenced the significant increase in robustness we observed in our evaluation. Given the complexity of the evaluation environment, this increase in non-linearity could have contributed to the increase in robustness, as we see that our relative robustness performance increased as the complexity of the dataset increased [51]. A better understanding of the relationship between the linearity of activation functions and the complexity of the data manifold on adversarial robustness is left for future work.

VII. Conclusion

The efforts regarding robust activation functions have often been pursued in isolation, leading to separate and parallel conclusions. As a result, we unified research efforts regarding robust activation functions through our parametric generalized gamma activation function that is non-monotonic, symmetric, bounded with finite support, smooth, and with relatively low curvature. We examined why and how consolidating these traits improved robustness against gradient-based attacks compared to the ReLU, hyperbolic tangent, swish, and SPLASH activation functions and how the different characteristics influenced the overall decision surface from the perspective of adversarial robustness. While this work shows an increase in adversarial robustness through the unification of characteristics explored disjointly by various research efforts, the complete list of characteristics desired in activation functions for adversarial robustness remains an open problem for future work. We provide non-linearity as an example for future work of a characteristic that is unclear of the magnitude of its impact on adversarial robustness [48].

References

[1] G. R. Machado, E. Silva, and R. R. Goldschmidt, "Adversarial machine learning in image classification: A survey toward the defender's perspective," *ACM Computing Surveys (CSUR)*, vol. 55, no. 1, pp. 1–38, 2021.

[2] A. Qayyum, M. Usama, J. Qadir, and A. Al-Fuqaha, "Securing connected & autonomous vehicles: Challenges posed by adversarial machine learning and the way forward," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 998–1026, 2020.

[3] K. D. Apostolidis and G. A. Papakostas, "A survey on adversarial deep learning robustness in medical image analysis," *Electronics*, vol. 10, no. 17, p. 2132, 2021.

[4] F. Aloraini, A. Javed, O. Rana, and P. Burnap, "Adversarial machine learning in iot from an insider point of view," *Journal of Information Security and Applications*, vol. 70, p. 103341, 2022.

[5] T. Bai, J. Luo, J. Zhao, B. Wen, and Q. Wang, "Recent advances in adversarial training for adversarial robustness," *International Joint Conference on Artificial Intelligence (IJCAI-21)*, 2021.

[6] J. Tack, S. Yu, J. Jeong, M. Kim, S. J. Hwang, and J. Shin, "Consistency regularization for adversarial robustness," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, pp. 8414–8422, 2022.

[7] S.-A. Rebuffi, S. Gowal, D. A. Calian, F. Stimberg, O. Wiles, and T. A. Mann, "Data augmentation can improve robustness," *Advances in Neural Information Processing Systems*, vol. 34, pp. 29935–29948, 2021.

[8] T. Chen, Z. Zhang, P. Wang, S. Balachandra, H. Ma, Z. Wang, and Z. Wang, "Sparsity winning twice: Better robust generalization from more efficient training," *arXiv preprint arXiv:2202.09844*, 2022.

[9] S.-M. Moosavi-Dezfooli, A. Fawzi, J. Uesato, and P. 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.

[10] M. Tavakoli, F. Agostinelli, and P. Baldi, "Splash: Learnable activation functions for improving accuracy and adversarial robustness," *Neural Networks*, vol. 140, pp. 1–12, 2021.

[11] A. Rozsa and T. E. Boult, "Improved adversarial robustness by reducing open space risk via tent activations," *arXiv preprint arXiv:1908.02435*, 2019.

[12] S. Dai, S. Mahloujifar, and P. Mittal, "Parameterizing activation functions for adversarial robustness," in *2022 IEEE Security and Privacy Workshops (SPW)*, pp. 80–87, IEEE, 2022.

[13] S. Fort, H. Hu, and B. Lakshminarayanan, "Deep ensembles: A loss landscape perspective," *arXiv preprint arXiv:1912.02757*, 2019.

[14] N. P. Baskerville, J. P. Keating, F. Mezzadri, and J. Najnudel, "The loss surfaces of neural networks with general activation functions," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2021, no. 6, p. 064001, 2021.

[15] F. Yu, C. Liu, Y. Wang, L. Zhao, and X. Chen, "Interpreting adversarial robustness: A view from decision surface in input space," *arXiv preprint arXiv:1810.00144*, 2018.

[16] I. Fursov, A. Zaytsev, N. Kluchnikov, A. Kravchenko, and E. Burnaev, "Gradient-based adversarial attacks on categorical sequence models via traversing an embedded world," in *Analysis of Images, Social Networks and Texts: 9th International Conference, AIST 2020, Skolkovo, Moscow, Russia, October 15–16, 2020, Revised Selected Papers 9*, pp. 356–368, Springer, 2021.

[17] N. Pitropakis, E. Panaousis, T. Giannetsos, E. Anastasiadis, and G. Loukas, "A taxonomy and survey of attacks against machine learning," *Computer Science Review*, vol. 34, p. 100199, 2019.

[18] P. Liu and G. Zheng, "Handling imbalanced data: Uncertainty-guided virtual adversarial training with batch nuclear-norm optimization for semi-supervised medical image classification," *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 7, pp. 2983–2994, 2022.

[19] S. Alemany and N. Pissinou, "The dilemma between data transformations and adversarial robustness for time series application systems," in *SafeAI at AAAI*, 2022.

[20] S. Kanai, M. Yamada, H. Takahashi, Y. Yamanaka, and Y. Ida, "Relationship between nonsmoothness in adversarial training, constraints of attacks, and flatness in the input space," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[21] L. Wu and Z. Zhu, "Towards understanding and improving the transferability of adversarial examples in deep neural networks," in *Asian Conference on Machine Learning*, pp. 837–850, PMLR, 2020.

[22] N. Boumal, *An introduction to optimization on smooth manifolds*. Cambridge University Press, 2023.

[23] H. Rangwani, S. K. Aithal, M. Mishra, A. Jain, and V. B. Radhakrishnan, "A closer look at smoothness in domain adversarial training," in *International Conference on Machine Learning*, pp. 18378–18399, PMLR, 2022.

[24] S. Kanai, M. Yamada, H. Takahashi, Y. Yamanaka, and Y. Ida, "Smoothness analysis of adversarial training," *arXiv preprint arXiv:2103.01400*, 2021.

[25] Y. Li, Y. Guo, Y. Xie, and Q. Wang, "A survey of defense methods against adversarial examples," in *2022 8th International Conference on Big Data and Information Analytics (BigDIA)*, pp. 453–460, IEEE, 2022.

[26] J. Xu, D. A. Yap, and V. U. Prabhu, "Understanding adversarial robustness through loss landscape geometries," in *Proc. of the International Conference on Machine Learning (ICML) Workshops*, p. 18, 2019.

[27] Q. Zhao and L. D. Griffin, "Suppressing the unusual: towards robust cnns using symmetric activation functions," *arXiv preprint arXiv:1603.05145*, 2016.

[28] L. Parisi, D. Neagu, R. Ma, and F. Campean, "Qrelu and m-qrelu: Two novel quantum activation functions to aid medical diagnostics," *arXiv preprint arXiv:2010.08031*, 2020.

[29] V. Singla, S. Singla, S. Feizi, and D. Jacobs, "Low curvature activations reduce overfitting in adversarial training," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16423–16433, 2021.

[30] R. Abdulhammed, H. Musafer, A. Alessa, M. Faezipour, and A. Abuzneid, "Features dimensionality reduction approaches for machine learning based network intrusion detection," *Electronics*, vol. 8, no. 3, p. 322, 2019.

[31] A. Athalye, N. Carlini, and D. 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.

[32] P. Sibi, S. A. Jones, and P. Siddarth, "Analysis of different activation functions using back propagation neural networks," *Journal of theoretical and applied information technology*, vol. 47, no. 3, pp. 1264–1268, 2013.

[33] D. Dũng *et al.*, "Deep relu neural networks in high-dimensional approximation," *Neural Networks*, vol. 142, pp. 619–635, 2021.

[34] H. Ide and T. Kurita, "Improvement of learning for cnn with relu activation by sparse regularization," in *2017 international joint conference on neural networks (IJCNN)*, pp. 2684–2691, IEEE, 2017.

[35] N. Liao, S. Wang, L. Xiang, N. Ye, S. Shao, and P. Chu, "Achieving adversarial robustness via sparsity," *Machine Learning*, pp. 1–27, 2022.

[36] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," *arXiv preprint arXiv:1710.05941*, 2017.

[37] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[38] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," 2009.

[39] S. Seong, Y. Lee, Y. Kee, D. Han, and J. Kim, "Towards flatter loss surface via nonmonotonic learning rate scheduling.," in *UAI*, pp. 1020–1030, 2018.

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

[41] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *stat*, vol. 1050, p. 20, 2015.

[42] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *International Conference on Learning Representations*, 2018.

[43] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 ieee symposium on security and privacy (sp)*, pp. 39–57, Ieee, 2017.

[44] M.-i. Nicolae, M. Sinn, M. N. Tran, B. Buesser, A. Rawat, M. Wistuba, V. Zantedeschi, N. B. Angel, B. Chen, H. Ludwig, *et al.*, "Adversarial robustness toolbox v1. 0.0," *arXiv*, 2019.

[45] R. Wang, R. Walters, and R. Yu, "Incorporating symmetry into deep dynamics models for improved generalization," *arXiv preprint arXiv:2002.03061*, 2020.

[46] F. Liu, M. Xu, G. Li, J. Pei, L. Shi, and R. Zhao, "Adversarial symmetric gans: Bridging adversarial samples and adversarial networks," *Neural Networks*, vol. 133, pp. 148–156, 2021.

[47] Y. Li, S. Cheng, H. Su, and J. Zhu, "Defense against adversarial attacks via controlling gradient leaking on embedded manifolds," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII 16*, pp. 753–769, Springer, 2020.

[48] C. Etmann, S. Lunz, P. Maass, and C. Schoenlieb, "On the connection between adversarial robustness and saliency map interpretability," in *International Conference on Machine Learning*, pp. 1823–1832, PMLR, 2019.

[49] C. Stephenson, A. Ganesh, Y. Hui, H. Tang, S. Chung, *et al.*, "On the geometry of generalization and memorization in deep neural networks," in *International Conference on Learning Representations*, 2020.

[50] S. Goldt, M. Mézard, F. Krzakala, and L. Zdeborová, "Modeling the influence of data structure on learning in neural networks: The hidden manifold model," *Physical Review X*, vol. 10, no. 4, p. 041044, 2020.

[51] O. F. Tuna, F. O. Catak, and M. T. Eskil, "Unreasonable effectiveness of last hidden layer activations for adversarial robustness," in *2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC)*, pp. 1098–1103, IEEE, 2022.