mathematics

MDPI

Article

Dynamic-Max-Value ReLU Functions for Adversarially Robust Machine Learning Models

Korn Sooksatra * and Pablo Rivas *

Department of Computer Science, Baylor University, Waco, TX 76706, USA
* Correspondence: korn_sooksatra1@baylor.edu (K.S.); pablo_rivas@baylor.edu (P.R.)

Abstract: The proliferation of deep learning has transformed artificial intelligence, demonstrating prowess in domains such as image recognition, natural language processing, and robotics. Nonetheless, deep learning models are susceptible to adversarial examples, well-crafted inputs that can induce erroneous predictions, particularly in safety-critical contexts. Researchers actively pursue countermeasures such as adversarial training and robust optimization to fortify model resilience. This vulnerability is notably accentuated by the ubiquitous utilization of ReLU functions in deep learning models. A previous study proposed an innovative solution to mitigate this vulnerability, presenting a capped ReLU function tailored to bolster neural network robustness against adversarial examples. However, the approach had a scalability problem. To address this limitation, a series of comprehensive experiments are undertaken across diverse datasets, and we introduce the dynamic-max-value ReLU function to address the scalability problem.

Keywords: machine learning; adversarial machine learning; robustness; trustworthy AI; adversarial examples

MSC: 68T07; 68Q32; 62H30; 65K10

Citation: Sooksatra, K.; Rivas, P. Dynamic-Max-Value ReLU Functions for Adversarially Robust Machine Learning Models. *Mathematics* **2024**, *12*, 3551. https://doi.org/10.3390/math12223551

Academic Editor: Ke-Lin Du

Received: 15 October 2024
Revised: 10 November 2024
Accepted: 11 November 2024
Published: 13 November 2024

1. Introduction

Over the past few years, the adoption of deep machine learning models across various sectors has significantly increased. This is attributed to their superior performance in numerous tasks, and some have outperformed human capabilities. These tasks span from medical diagnosis to autonomous driving, where the accuracy and reliability of machine learning predictions are crucial. In the context of autonomous vehicles, for instance, the robustness of these systems is non-negotiable, as any failure could potentially endanger lives.

However, deep learning models exhibit a critical vulnerability to adversarial examples, i.e., subtle and deliberately engineered modifications to input data crafted to mislead the model into making erroneous decisions. Figure 1 illustrates an adversarial example that can fool an image classifier into predicting the image as a cat instead of as a dog. This susceptibility was first identified and discussed in seminal papers [1,2], highlighting significant challenges in deploying these models in environments demanding high security and reliability.

In our work, we show that the Static-Max-Value ReLU (S-ReLU) function designed in [3] is theoretically more robust than a traditional ReLU function. However, it does not work well on large datasets. Based on that analysis, we introduce the Dynamic-Max-Value ReLU (D-ReLU) function, an advanced activation function designed to dynamically adjust based on input data. This innovation enhances the model's robustness, particularly when applied to larger and more complex datasets. By tailoring the activation mechanism to the specific characteristics of the input, D-ReLU aims to improve the overall performance of deep learning models.

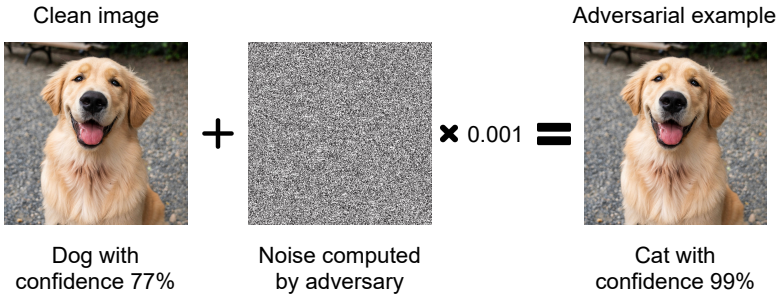


Figure 1. Adversarial example that misleads an image classifier to predict the image as a cat.

Additionally, we explore the integration of D-ReLU into state-of-the-art pre-trained deep learning models. Through this integration, we demonstrate how modifications to traditional ReLU functions, alongside the addition of dense layers, can significantly enhance model security and reliability. This approach not only modernizes the activation function but also contributes to the stability and performance of established architectures.

To validate the effectiveness of D-ReLU, we conduct comprehensive experiments across large datasets, such as CIFAR-10, CIFAR-100, and TinyImagenet. These experiments aim to evaluate the practical performance and robustness improvements that D-ReLU can offer. The findings from these studies underscore the potential of D-ReLU for safer public deployment of deep learning models, making a compelling case for its adoption in various applications.

The rest of this paper is organized as follows. Section 2 discusses works with the same goals as our approach. Section 3 mentions and further analyzes the work inspiring us to propose our approach. Section 4 defines our approach. Section 5 shows the setups of our experiments. Section 6 details experimental applications of our approach and baselines against white-box attacks. Section 7 details experimental applications of our approach and baselines against white-box attacks. Section 8 reports on experiments with the models trained on augmented datasets against adversarial attacks. Section 9 discusses how much our approach generalizes in several perturbation bounds. Section 10 explains the limitations of our approach. Section 11 demonstrates the broader impact of our approach. Section 12 concludes everything and discusses future works.

2. Related Works

Addressing the vulnerabilities posed by adversarial examples has led to a plethora of research endeavors [2,4–9]. Among the proposed solutions, adversarial training has emerged as a foremost strategy due to its relatively straightforward implementation and proven effectiveness [7]. This approach involves training the model on a dataset supplemented with adversarially modified examples, thereby improving the model's resilience to similar attacks. However, the technique significantly extends the training duration and computational demands.

Moreover, integrating autoencoders and generative adversarial networks (GANs) has been explored to preprocess and potentially cleanse adversarial perturbations from inputs [10,11]. These methods aim to improve the robustness of machine learning models by denoising or altering the input data before the model processes them. Figure 2 demonstrates the autoencoder method, which preprocesses an input to ensure that the classifier or target model receives a clean input. This autoencoder was specifically trained to denoise adversarial examples, effectively reducing the impact of adversarial perturbations. However, these solutions necessitate additional model training and are challenged by the complexity of handling large-scale data. The extra computational overhead and the need for extensive training make these methods less feasible for real-time applications and large datasets.

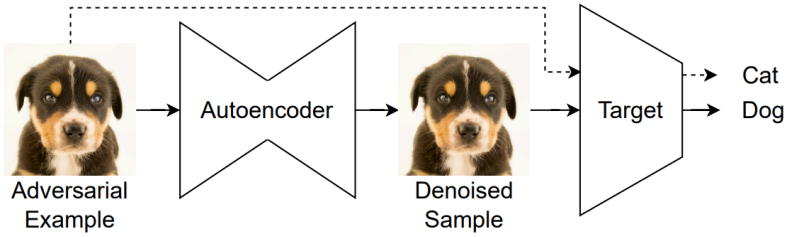


Figure 2. Denoised autoencoder for preprocessing of an adversarial example to create a clean/denoised sample. The solid line is the process with the autoencoder, and the dashed line is the process without the autoencoder.

In addition, some strategies have focused on leveraging the outputs from machine learning models to promote robustness. One notable approach is randomized smoothing [12], which involves injecting random noise, such as Gaussian noise, into the input and generating multiple noisy versions of the input. Each version is then passed through the machine learning model, and the final prediction is determined by a majority vote among the predictions from the noisy inputs. While randomized smoothing provides a form of certified robustness, it has significant drawbacks. The method requires multiple passes through the model for each prediction, which is computationally expensive and impractical for real-time systems, where quick responses are essential. Figure 3 illustrates the randomized smoothing method. In this example, the system generates four predictions: three predict the input as a dog, and one predicts it as a cat. Based on the majority vote, the final output of the system is a dog.

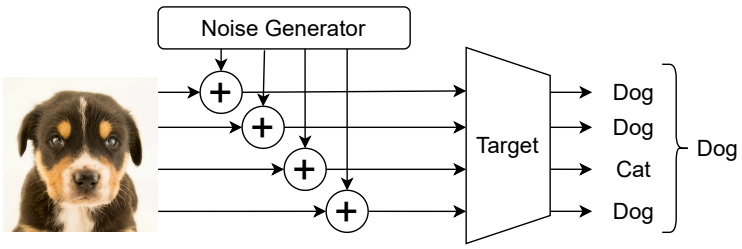


Figure 3. Randomized smoothing method, where the most common predictions are picked as the output. In this example, four noises are generated by the noise generator.

Another method, defensive distillation, aims to reduce a model's sensitivity to input variations by training the model to output softened probabilities rather than hard classifications [13]. Despite its initial promise, defensive distillation is vulnerable to more sophisticated adversarial attacks, as demonstrated by Carlini and Wagner [14]. This finding indicates that while defensive distillation can provide some level of robustness, it is not a comprehensive solution and only offers absolute protection against some types of adversarial techniques.

Many works have also attempted to create detectors for adversarial examples, aiming to filter out adversarial inputs before they reach the machine learning model [15–20]. These detectors can identify potentially malicious inputs and prevent them from affecting the model's predictions. However, these approaches do not inherently improve the robustness of the underlying machine learning models. Furthermore, some detector-based methods rely on additional machine learning models, which can be vulnerable to adversarial attacks, allowing attackers to bypass the detectors and compromise the target models. Figure 4 depicts this technique. Samples detected as adversarial examples are ignored. Another drawback is that there will be no input for the machine learning model if there are only adversarial examples in the real world.

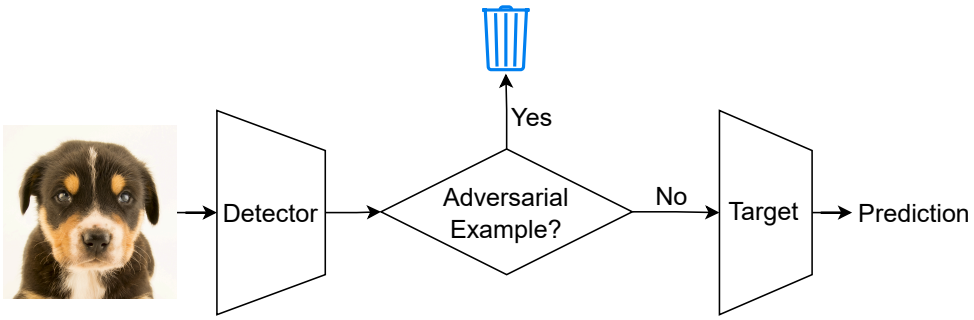


Figure 4. Adversarial example detection technique where the detected samples are thrown away.

All the techniques mentioned above focus on preprocessing, post processing, or augmenting the inputs and outputs rather than directly modifying the models' architectures. However, the architecture of the models—mainly the activation functions—significantly contributes to their vulnerability to adversarial examples. As illustrated in Figure 1, these tiny perturbations are imperceptible to the human eye. Despite this, certain activation functions, such as ReLU, enable these perturbations to amplify as they propagate through the layers of machine learning models. Ultimately, this can lead to changes in the output-layer values, consequently affecting the final prediction.

While some research [21–28] has aimed to customize the models' architectures, only a few works have specifically targeted the customization of activation functions. One such effort by [21] involved quantizing activation functions, which can significantly reduce the precision of activations and potentially degrade the model's performance and robustness. Another example is the ReLU6 activation function used in MobilenetV2 [29], which caps the activation values at 6. Although ReLU6 was introduced to improve robustness, its potential must be further explored to enhance robustness against adversarial attacks.

Amidst these challenges, we observe a potential opportunity with respect to conventional solutions concerning the activation functions within deep learning architectures—in particular, the ReLU function, which is known to contribute to vulnerabilities against adversarial examples [2]. A ReLU function is formulated as $\max(x, 0)$, where x is an input and $\max(\cdot, \cdot)$ outputs the maximum value between two parameters. Our research aims to directly address this by enhancing the design of ReLU functions to improve model robustness without compromising accuracy.

3. Static-Max-ReLU Function

This novel activation function is termed the static-max-value ReLU function (S-ReLU) and is defined as follows:

$$\text{S-ReLU}(x, m) = \max(0, \min(m, x)),$$

where x represents the incoming input and m is a predefined maximum value. Figure 5 shows an example of this activation function, where the max value is 2. We can see that the function is capped after the input is 2. Theoretical analyses are presented to demonstrate the enhanced robustness of this proposed function compared to a general ReLU function. Furthermore, empirical experiments are detailed in the subsequent sections to empirically validate its improved robustness.

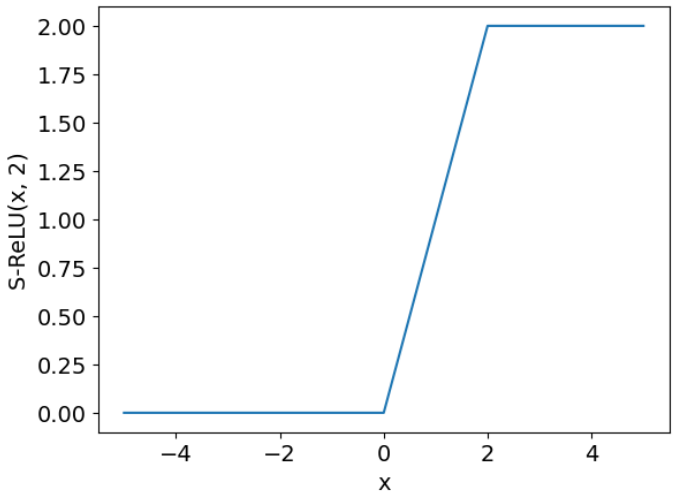


Figure 5. An example of S-ReLU with a max value of 2.

3.1. Theoretical Analysis

We previously introduced S-ReLU. Next, we aim to theoretically demonstrate how S-ReLU can neutralize adversarial perturbations at each layer in this section.

Theorem 1. *The outputs of S-ReLU functions always have an equal number of perturbations or fewer relative to the outputs of ReLU functions, given the same inputs. Note that perturbations are the additional noises affected by the corrupted input (also see Appendix A).*

The utilization of the static-max-value ReLU function (S-ReLU) is likely associated with a reduction in the Lipschitz constant, denoted as K. This observation is substantiated by the findings presented in Theorem 1, which indicate a diminished discrepancy between the outputs of a layer when processing a clean sample and an adversarial example, especially when contrasted with the behavior of the standard ReLU activation function. The Lipschitz inequality is expressed as

$$d_Y(f(x), f(x^*)) \leq K \cdot d_X(x, x^*),$$

where x is a clean sample, x^* is its adversarial example, $f(\cdot)$ is a classifier, $d_X(\cdot, \cdot)$ is a distance function (e.g., L_2 norm and L_∞ norm) for an input, and $d_Y(\cdot, \cdot)$ is a distance function (e.g., L_2 norm) for an output. The consequential reduction in the Lipschitz constant, a consequence of employing S-ReLU, signifies an enhancement in the model's robustness, as a lower Lipschitz constant is indicative of reduced sensitivity to input perturbations and, consequently, increased resilience against adversarial examples.

Next, we theoretically show how the max value (denoted by m) affects the amount of adversarial perturbations in a layer.

Corollary 1. *When the max value (m) of S-ReLU in a layer reduces, the layer's outputs of clean samples and adversarial examples are closer (also see Appendix B).*

According to Corollary 1, we can reduce the max values of S-ReLU to reduce the Lipschitz constant and eventually improve robustness. However, this technique may harm the overall performance if the max values are too low.

3.2. Limitations

In this section, we discuss the limitations of S-ReLU. While S-ReLU successfully enhances adversarial robustness in MNIST classifiers, its performance falters when applied to more extensive datasets like CIFAR-10. The challenge arises from the substantial number of layers and zero gradients, leading to what is commonly known as the gradient vanishing

problem. The upcoming sections explain a compelling solution to address and overcome this issue, revolutionizing the capability of classifiers on larger datasets.

4. Dynamic-Max-ReLU Functions

Sooksatra et al. [3] demonstrated the effectiveness of S-ReLU in enhancing both model performance and adversarial robustness. They conducted a series of experiments that showcased improvements in resistance to adversarial attacks facilitated by the S-ReLU function. We also showed the theoretical results in the previous section. However, we observed challenges when applying S-ReLU to larger datasets beyond MNIST, primarily due to issues related to gradient vanishing.

To address these challenges, this section introduces a new variant, the dynamic-max-value ReLU function (D-ReLU). This modified function aims to retain the advantages of S-ReLU while mitigating its limitations on larger datasets. This approach uses the same activation functions as S-ReLU. However, the max values (i.e., m) of those functions are learnable. Therefore, at first, we set those values to be high, then try to minimize them during training to improve the robustness such that the optimizer can adjust the models with low max values. We minimize the max values because Table A1 shows that low max values (i.e., m) lead to small output differences and improve robustness. Therefore, the loss function can be formulated as

$$l(F(x,\theta),y) + \lambda \sum_i m_i^2, \tag{1}$$

where $F(x,\theta)$ is a classifier, x is an input, θ represents the parameters of F, y is the true label, m_i is the max value of neuron i with D-ReLU as its activation function, and λ balances the model's performance and adversarial robustness. Note that the value of m_i is the number of neurons whose activation functions are D-ReLU. Next, we illustrate how D-ReLU can enhance adversarial robustness through a series of experiments. Before presenting our findings, we first describe the experimental setup.

5. Experimental Setup

In this section, we provide a comprehensive breakdown of the methodologies and resources utilized to configure and conduct our experimental studies. The components detailed here are crucial for replicating our results and understanding the efficacy of our proposed modifications in terms of model robustness.

First, we discuss the datasets employed in our experiments. These datasets were carefully selected to cover a variety of scenarios and complexity levels, which helps in testing the resilience of our modified models across different data distributions and task complexities.

Secondly, we elaborate on the specific training details, which include the configuration of the machine learning models, the choice of hyperparameters, and the training procedures we adopted.

Next, we delve into the robustness evaluations. Here, we define the metrics and methodologies used to assess the robustness of the models against adversarial attacks. This includes a description of how adversarial examples were generated and the criteria used to evaluate the model's performance in the face of such perturbations.

Finally, we outline the baselines for comparison. This includes a discussion of the existing models and techniques against which our proposed modifications were benchmarked. Describing these baselines provides context for the improvements our research introduces and furnishes a clear contrast to demonstrate the incremental gains in robustness attributed to our enhancements.

Each of these elements plays a vital role in shaping the experimental design and is critical for assessing the practical impact of our research in enhancing the robustness of deep learning models against sophisticated adversarial threats.

5.1. Datasets

We used four datasets in this experiment: MNIST [30], CIFAR10 [31], CIFAR100 [31], and TinyImagenet [32]. MNIST consists of 60,000 training images and 10,000 testing images, each 28 × 28 pixels, representing handwritten digits from 0 to 9. Figure 6 shows some examples from this dataset.

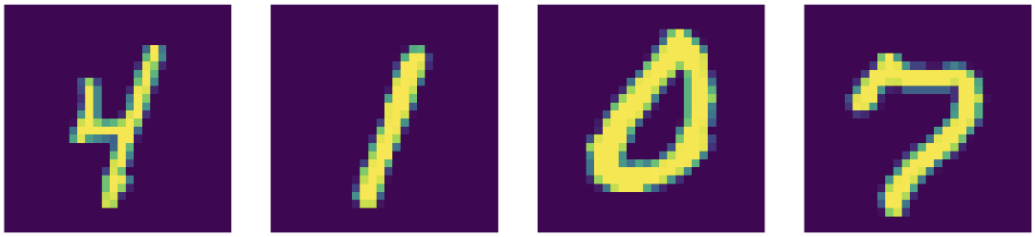


Figure 6. Examples of the MNIST dataset.

CIFAR10 is a dataset commonly used for machine learning and computer vision tasks. CIFAR-10 consists of 60,000 32 × 32 color images in 10 different classes, with each class representing a distinct object or animal category. The dataset is divided into 50,000 training images and 10,000 testing images. It is widely used as a benchmark for developing and evaluating image classification algorithms and models. Figure 7 shows some examples of this dataset.

Figure 7. Examples of the CIFAR10 dataset.

The CIFAR100 dataset is a collection of 60,000 32 × 32 color images across 100 different classes, with each class containing 600 images. It serves as a benchmark for image classification tasks, where each image belongs to one of the 100 fine-grained object classes. This dataset is commonly used for evaluating machine learning algorithms and models due to its diverse set of classes and relatively small image size. Figure 8 shows some examples of this dataset.

Figure 8. Examples of the CIFAR100 dataset.

TinyImagenet is a subset of the large-scale Imagenet dataset designed for the training of deep neural networks with smaller computational resources. It consists of 200 diverse classes, with each class having 100,000 training images and 10,000 test images. Each image has dimensions of 64 × 64 pixels, representing a wide range of object categories, making it a useful dataset for tasks like classification, detection, and segmentation. TinyImagenet serves as a more manageable alternative to the full Imagenet dataset for researchers and

practitioners working on computer vision tasks. We partitioned the training set using an 80/20 ratio for validation. Figure 9 shows some examples of this dataset.

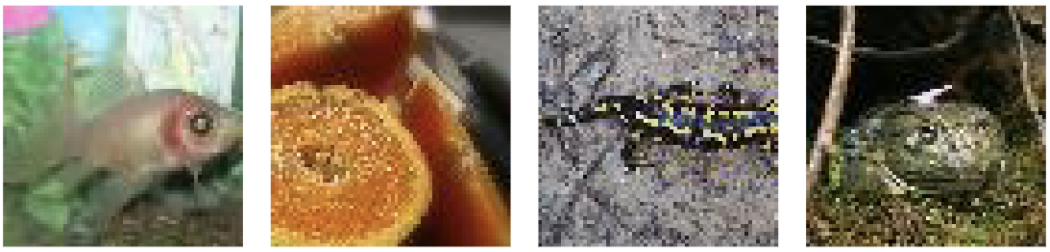


Figure 9. Examples of the TinyImagenet dataset.

5.2. Training Details

We used Tensorflow for this implementation. The optimization process employed the Adam optimizer [33] with the initial learning rate set to 10^{-3}. Additionally, we implemented the ReduceLROnPlateau callback with a decay factor of 0.5 and a patience of 5, as well as the EarlyStopping callback with patience of 10 based on the validation loss. The ReduceLROnPlateau callback reduces the learning rate by multiplying it by its decay factor when the validation loss does not improve for the patience epochs. The EarlyStopping callback stops the training when the validation loss does not improve for the patience epochs. The maximum number of epochs for the training procedure was set to 2000. We conducted three independent training sessions for each model type. All subsequent results presented in the following sections represent the average performance obtained from these three trained models.

We also added a dense layer before the output layer. Incorporating a dense layer before the output layer is motivated by findings from the experiment conducted in [3]. The study demonstrated that employing S-ReLU in the last hidden layer yields superior results compared to its placement in earlier layers. This layer's activation function is D-ReLU for our approach, as shown in Figure 10, while it is a general ReLU for other approaches.

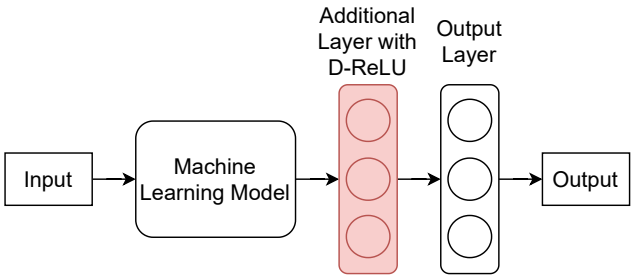


Figure 10. Architecture of our approach with an added layer (in red) with D-ReLU before the output layer.

5.3. Adversarial Attacks

We employed diverse adversarial attack strategies to compute the robust accuracy for trained targeted models by using the test samples. The selected attacks encompass the following methodologies:

- Fast Gradient Sign Method (FGSM) [2]: This attack creates adversarial examples by perturbing input data in the direction that maximizes the model's loss, utilizing the sign of the gradients and a small constant.
- Project Gradient Descent (PGD) [7]: This attack is an iterative approach, repeatedly updating the input by taking small steps in the gradient direction and projecting the result back into a small neighborhood around the original data. While FGSM is computationally less intensive and involves a single step, PGD is generally more

effective and robust, requiring multiple iterations but producing adversarial examples that are harder to defend against.

- Auto Projected Gradient Descent with DLR (APGD_DLR) [34]: This variant of APGD_CE maintains the same underlying principles as APGD_CE but employs the Difference of Logit Ratio Loss (DLR) [34] as the loss function.
- Carlini and Wagner Attack with L2 Norm (CW_L2) [6]: Diverging from the optimization-based approach of the preceding attacks, CW_L2 is characterized by its slower adversarial example discovery process. However, its potency in generating robust adversarial examples is noteworthy. It directly minimizes the difference between clean samples and adversarial examples with the L2 norm and maximizes the misclassification confidence as well.
- Square [35]: This is a black-box attack that utilizes random initialization with vertical stripes to perturb images within a specified range. By focusing on sparse updates grouped in a square pattern, the attack strategically alters the input, aiming to induce subtle yet significant changes in image components. This method leverages the sensitivity of convolutional networks to high-frequency perturbations and is designed to generate successful perturbations within a limited radius, ensuring distinct differences relative to the original image. By strategically manipulating color channels and employing sparse updates, the attack aims to maximize perturbation impact while adhering to image constraints and network sensitivities.

5.4. SOTA Methods for Robustness

To justify our approach's novelty, we also compared it to state-of-the-art methods for adversarial robustness. We selected the following popular and effective methods:

- Adversarial Training [7]: This method retrains a model with adversarial examples after its successful natural training. We retrained the models for 10 epochs.
- TRADES [36]: This method balances the performance and robustness of a model by customizing the loss function. The loss function consists of two parts. The first part increases the performance, and the other part improves the robustness by computing the difference between the output distributions between the clean samples and their adversarial counterparts. Please be aware that this method utilizes a parameter denoted as β to strike a balance between performance and robustness. We adopted the same values of β as those employed by the original authors—specifically, $\beta = 1$ and 6.

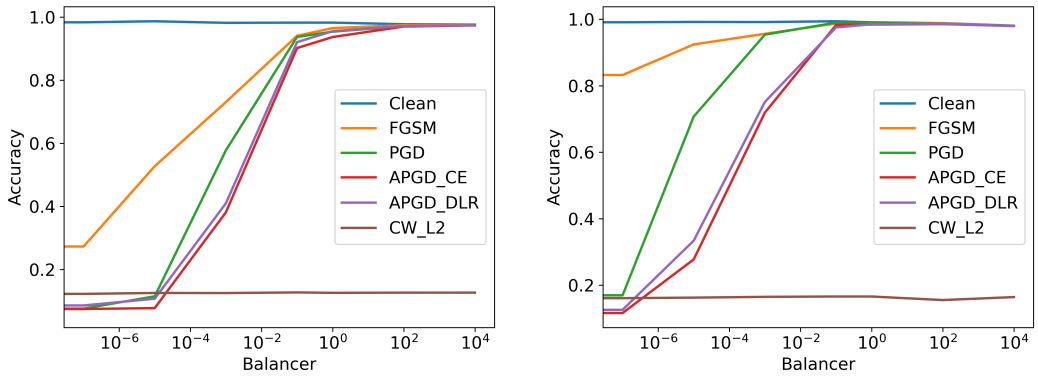
We used PGD to generate adversarial examples for all the mentioned methods.

6. White-Box Attack Experiments

6.1. Experimental Results for MNIST

We created two models for the MNIST dataset. The first one is a two-hidden-layer dense network, and the other one is a shallow convolutional network. These networks are enough to evaluate the MNIST dataset. We set the perturbation bound to 0.1 for FGSM, PGD, PGD_CE, and PGD_DLR. We also set the perturbation bound to 18 for CW_L2.

The outcomes of tuning the balancer, denoted as λ in (1), are illustrated in Figure 11. Note that at a balancer value of zero, the models were naturally trained, and they were not robust against attacks at all. Through experimentation on both a dense network and a shallow CNN, it was observed that elevating the balancer led to increased accuracies on adversarial examples generated by FGSM, PGD, APGD_CE, and APGD_DLR. Interestingly, this improvement in adversarial accuracy occurred while the accuracy on clean samples remained relatively stable. This outcome aligns with our expectations. However, in the case of adversarial examples generated by CW_L2, the accuracy did not exhibit a similar increase. This anomaly can be attributed to the strength of the CW_L2 attack, where the applied perturbation may remain consistent across all samples.



(**a**) Dense (**b**) Shallow CNN

Figure 11. Accuracy of two types of networks on clean MNIST and adversarial examples when adding a dense layer with a D-ReLU function before the output layer.

Table 1 presents the performance (accuracy on clean samples) and robustness (accuracy on adversarial examples) achieved by training models using both state-of-the-art methods and our proposed approach. We carefully selected the optimal trade-off between performance and robustness for our approach, with the corresponding balancer values detailed in the table. Notably, our approach outperforms other methods across various scenarios, except for the accuracy of the dense model on both clean samples and adversarial examples generated by CW_L2. Importantly, our method achieves this superior performance without the need to compute adversarial examples during the training process. This observation underscores the efficacy of our approach in endowing machine learning models with adversarial robustness without compromising overall performance.

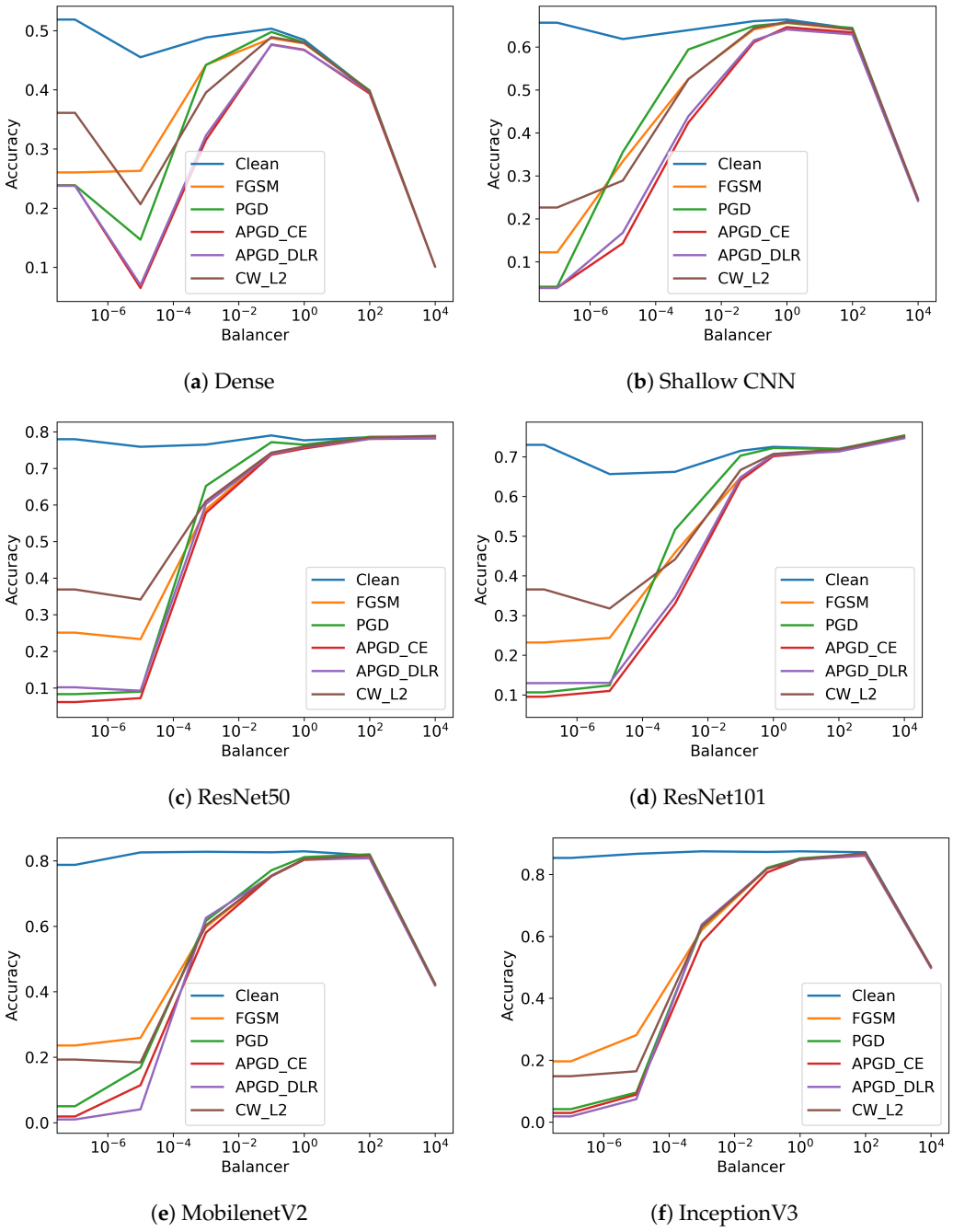
Table 1. Accuracy metrics for dense networks and shallow CNNs under various robust training schemes, evaluating them on both clean samples and adversarial examples generated by different attacks on the MNIST dataset. Note that AP_{CE} is $APGD_{CE}$, AP_{DLR} is $APGD_{DLR}$, the accuracy metrics in bold are the highest in a specific model among the different training methods, the numbers in parentheses are the ranks for training methods under an architecture, TRADES-k indicates the TRADES approach with $\beta = k$, and D-ReLU-k represents the D-ReLU approach with $m = k$.

Model	**Training**	**Clean** %	**FGSM** %	**PGD** %	$\mathbf{AP_{CE}}$ %	$\mathbf{AP_{DLR}}$ %	$\mathbf{CW_{L2}}$ %
Dense	AT	98.10 (1)	89.77 (4)	87.70 (4)	87.63 (4)	87.47 (4)	12.57 (4)
	TRADES-1	98.07 (2)	93.03 (2)	90.87 (2)	90.97 (2)	90.83 (2)	**16.50** (1)
	TRADES-6	96.20 (4)	91.40 (3)	89.53 (3)	89.57 (3)	89.13 (3)	12.83 (2)
	D-ReLU-10^2	97.77 (3)	**97.47** (1)	**97.10** (1)	**96.93** (1)	**97.03** (1)	12.63 (3)
Shallow CNN	AT	99.20 (2)	96.77 (3)	95.83 (3)	95.70 (3)	95.73 (3)	16.47 (2)
	TRADES-1	98.90 (3)	96.93 (2)	96.77 (2)	96.60 (2)	96.67 (2)	13.87 (4)
	TRADES-6	98.17 (4)	96.47 (4)	95.30 (4)	95.03 (4)	95.03 (4)	16.23 (3)
	D-ReLU-10^{-1}	**99.40** (1)	**98.73** (1)	**99.00** (1)	**98.30** (1)	**98.10** (1)	**16.60** (1)

6.2. Experimental Results for CIFAR10

We trained six types of models: two-hidden-layer dense networks, shallow convolutional neural networks (CNN), ResNet50 [37], ResNet101 [37], MobilenetV2 [29], and InceptionV3 [38]. We set the perturbation bounds to 0.01 for FGSM, PGD, APGD_CE, and APGE_DLR. Moreover, the bound for CW_L2 was set to 18.

Figure 12 provides a detailed visualization of the performance outcomes for various models that employ different balancer values under multiple adversarial attack scenarios. This figure enables a comparative analysis, particularly focusing on how these models withstand adversarial perturbations when adjusted with varying balancer levels.



(**a**) Dense (**b**) Shallow CNN

(**c**) ResNet50 (**d**) ResNet101

(**e**) MobilenetV2 (**f**) InceptionV3

Figure 12. Accuracy of several types of networks on clean CIFAR10 and adversarial examples when adding a dense layer with a D-ReLU function before the output layer.

Consistent with our prior observations on the MNIST dataset, we noted a similar trend in the CIFAR-10 dataset. Specifically, as the balancer values increase, there is a noticeable enhancement in robustness against several attacks. This pattern aligns with our expectations and demonstrates that carefully calibrated balancer values can significantly improve a model's resistance to certain types of adversarial attacks. However, it is important to highlight that while higher balancer values enhance robustness, there is a threshold beyond which further increases can negatively impact overall model performance. This suggests a trade-off where excessively high balancer values may lead to the deminishment of accuracy or other performance metrics under standard conditions.

In light of these findings, the D-ReLU mechanism appears to be particularly effective. For medium-sized datasets such as CIFAR10 and advanced models including ResNet, Mobilenet, and Inception, D-ReLU strikes a balance that optimizes robustness without

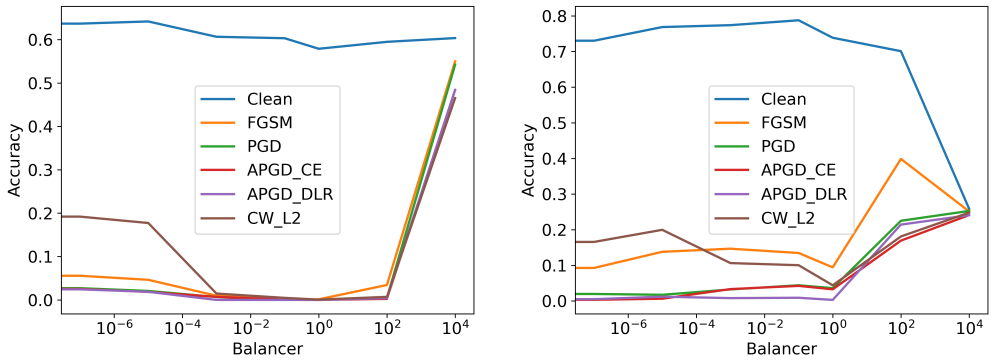
excessively compromising overall performance. This makes D-ReLU a promising choice for practitioners looking to enhance model robustness in practical applications.

The implications of these results are multifaceted. First, they underscore the importance of balancing robustness and performance. While enhancing defense mechanisms against adversarial attacks is crucial, maintaining high levels of accuracy and performance in non-adversarial scenarios is equally important. This balance ensures that the models remain useful and effective in real-world applications where both adversarial and benign inputs are encountered.

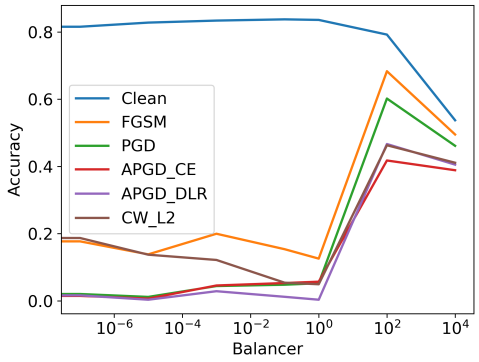
Secondly, the trend observed with escalating balancer values offers insights into the tuning process for adversarial robustness. It suggests that there is a critical balancer value range that optimizes defense mechanisms without significantly degrading the model's general performance. Identifying this optimal range can guide the development of more resilient machine learning systems.

Furthermore, the suitability of D-ReLU for state-of-the-art models such as ResNet, Mobilenet, and Inception indicates its potential for broader adoption. These models are widely used in various applications due to their performance and efficiency. Enhancing their robustness with D-ReLU can make them more reliable in adversarial settings, thereby extending their applicability in security-sensitive domains such as autonomous driving, medical imaging, and financial forecasting.

We also experimented with placing the additional convolutional layer with D-ReLU after the input layer instead of incorporating it in the dense layer before the output layer. Figure 13 presents the outcomes, illustrating the impact on several CNN architectures when the D-ReLU layer is added at the beginning of the network.



(**a**) Shallow CNN (**b**) MobilenetV2



(**c**) InceptionV3

Figure 13. Accuracy of several types of CNNs on clean CIFAR10 and adversarial examples when adding a convolutional layer with a D-ReLU function after the input layer.

The results indicate that positioning the D-ReLU layer early in the network does not yield the same level of effectiveness as when placed in deeper layers. For the Shallow CNN (Figure 13a), MobilenetV2 (Figure 13b), and InceptionV3 (Figure 13c), there is a notable decline in adversarial robustness across different attack types (FGSM, PGD, APGD_CE, APGD_DLR, and CW_L2) as compared to when the D-ReLU layer is situated deeper in the network. This trend suggests that the D-ReLU function, when applied later in the model, significantly enhances the model's ability to withstand adversarial attacks while maintaining high accuracy on clean samples.

The implications of these findings are significant for the design of robust neural network architectures. Incorporating D-ReLU in deeper layers allows the network to better leverage its properties for to improve adversarial robustness. This highlights the importance of strategic layer placement within CNNs, particularly for applications requiring high resilience to adversarial perturbations without compromising performance on clean data.

Table 2 provides a comprehensive comparison of accuracy metrics and rankings for various robust training schemes applied to different models on the CIFAR10 dataset. The table reveals that D-ReLU consistently achieves an optimal balance between performance on clean samples and robustness against adversarial attacks, particularly excelling in the context of deep networks like ResNet and InceptionV3.

Table 2. Accuracy metrics for multiple types of networks under various robust training schemes, evaluating them on both clean samples and adversarial examples generated by different adversarial attacks on the CIFAR10 dataset. Note that AP_{CE} is $APGD_{CE}$, AP_{DLR} is $APGD_{DLR}$, the accuracy metrics in bold are the highest in a specific model among the different training methods, the numbers in parentheses are the ranks for training methods under an architecture, TRADES-k indicates the TRADES approach with $\beta = k$, and D-ReLU-k represents the D-ReLU approach with $m = k$.

Model	Training	Clean %	FGSM %	PGD %	AP_{CE} %	AP_{DLR} %	CW_{L2} %
Dense	AT	**52.33** (1)	34.20 (2)	32.83 (2)	32.73 (2)	31.80 (2)	40.10 (2)
	TRADES-1	52.32 (2)	29.97 (3)	29.23 (3)	29.20 (3)	28.37 (3)	38.67 (3)
	TRADES-6	51.30 (4)	**37.00** (1)	**36.53** (1)	**36.50** (1)	**34.57** (1)	**42.30** (1)
	D-ReLU-10^{-7}	51.87 (3)	26.03 (4)	23.87 (4)	23.80 (4)	23.77 (4)	36.10 (4)
Shallow CNN	AT	67.13 (2)	42.83 (2)	40.07 (2)	39.90 (2)	38.37 (2)	50.67 (2)
	TRADES-1	**67.37** (1)	38.83 (4)	35.93 (4)	35.97 (4)	34.13 (4)	48.60 (4)
	TRADES-6	63.47 (4)	46.13 (3)	44.80 (3)	44.80 (3)	42.67 (3)	51.67 (3)
	D-ReLU-10^{0}	66.37 (3)	**65.60** (1)	**65.60** (1)	**64.60** (1)	**64.07** (1)	**65.83** (1)
ResNet50	AT	78.20 (2)	54.77 (2)	49.37 (2)	48.90 (2)	49.97 (2)	63.00 (2)
	TRADES-1	75.63 (4)	52.12 (4)	40.77 (4)	39.87 (4)	40.20 (4)	56.43 (4)
	TRADES-6	71.63 (3)	54.20 (3)	50.90 (3)	50.40 (3)	48.23 (3)	57.63 (3)
	D-ReLU-10^{4}	**78.87** (1)	**78.83** (1)	**78.73** (1)	**78.20** (1)	**78.40** (1)	**78.87** (1)
ResNet101	AT	68.90 (3)	44.90 (4)	40.33 (2)	39.43 (2)	38.27 (2)	49.30 (3)
	TRADES-1	74.60 (1)	47.07 (2)	32.87 (4)	31.17 (4)	31.37 (4)	51.40 (2)
	TRADES-6	66.67 (4)	45.43 (3)	39.80 (3)	39.17 (3)	35.93 (3)	47.67 (4)
	D-ReLU-10^{4}	**75.10** (2)	**75.03** (1)	**75.37** (1)	**74.73** (1)	**74.67** (1)	**75.10** (1)
MobilenetV2	AT	77.97 (2)	46.50 (2)	32.93 (4)	30.73 (4)	32.10 (4)	51.80 (2)
	TRADES-1	73.13 (4)	46.23 (3)	31.00 (3)	28.87 (3)	28.77 (3)	49.37 (4)
	TRADES-6	68.40 (3)	48.80 (2)	43.23 (2)	43.03 (2)	40.80 (2)	51.13 (3)
	D-ReLU-10^{2}	**81.67** (1)	**81.57** (1)	**82.00** (1)	**80.87** (1)	**80.77** (1)	**81.67** (1)
InceptionV3	AT	84.60 (2)	64.27 (2)	58.80 (2)	58.30 (2)	59.33 (2)	66.47 (2)
	TRADES-1	82.53 (3)	62.30 (3)	52.67 (4)	51.90 (4)	51.87 (4)	62.40 (4)
	TRADES-6	76.97 (4)	61.97 (4)	58.00 (3)	57.80 (3)	56.03 (3)	62.10 (3)
	D-ReLU-10^{2}	**87.17** (1)	**86.70** (1)	**86.57** (1)	**86.13** (1)	**86.23** (1)	**86.83** (1)

Interestingly, while TRADES with $\beta = 6$ demonstrated superior robustness for the dense network, it did so at the expense of performance on clean samples. In contrast, our D-ReLU approach significantly outperformed other methods in generalizing to adversarial examples, and it did so without the need to compute adversarial examples during training. This characteristic is particularly advantageous, as it simplifies the training process and reduces computational overhead.

Moreover, D-ReLU's ability to maintain high performance on clean samples is noteworthy. Unlike other robust training schemes that often sacrifice accuracy on clean data to gain adversarial robustness, D-ReLU preserves the integrity of clean sample performance, making it a highly efficient and practical approach for enhancing model robustness without compromising overall accuracy. This makes D-ReLU a highly effective method for deploying robust models in real-world scenarios where maintaining high accuracy on clean data is crucial.

Additionally, we performed an ANOVA test and obtained an F score of 17.4, surpassing the critical value of 3.92 at $\alpha = 0.01$. Given that the F score was significantly higher than the critical value, we reject the null hypothesis and conclude that there are significant differences among the approaches with 99% confidence. Moreover, considering the average accuracy, it is evident that D-ReLU significantly enhances model robustness.

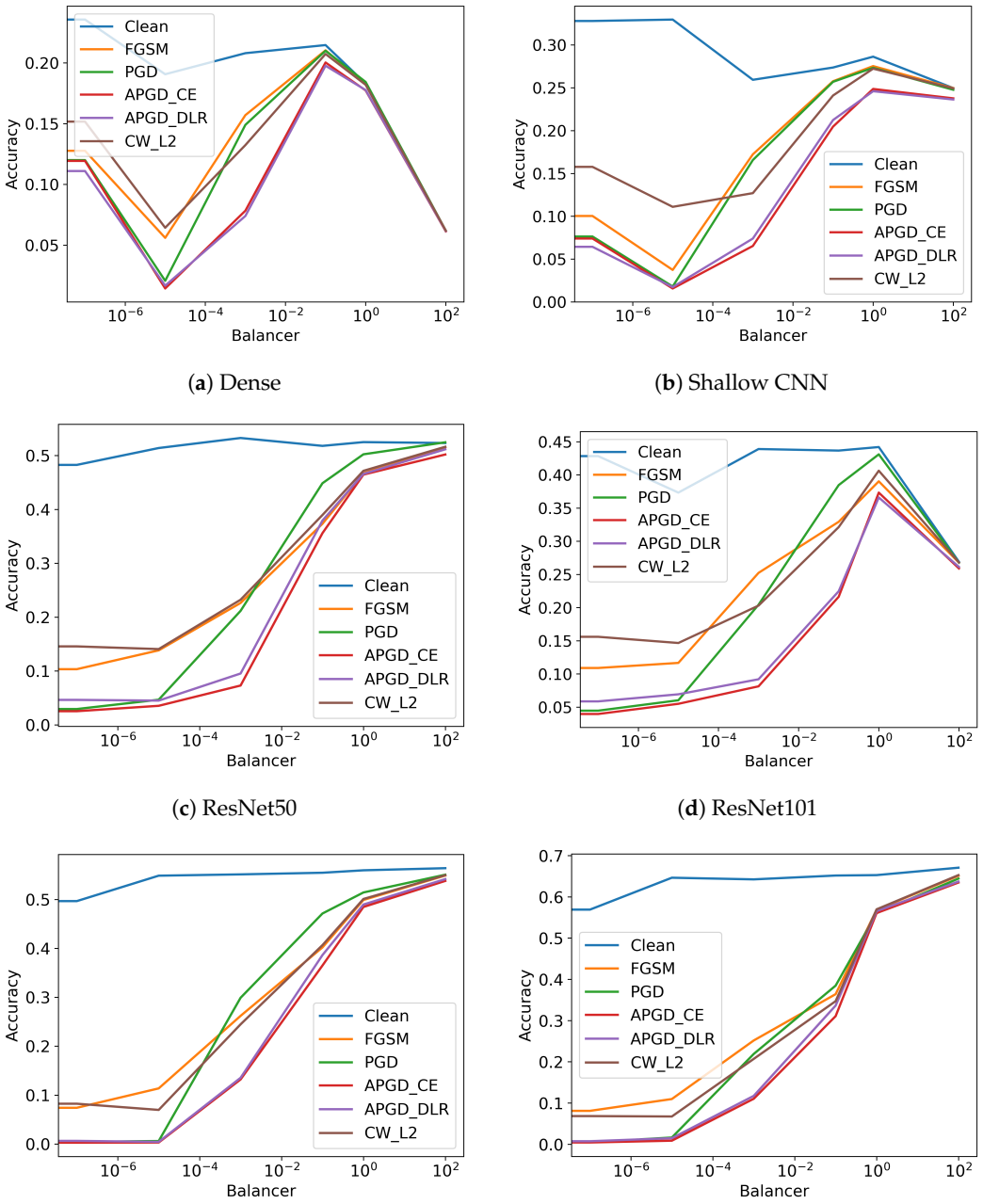
Furthermore, we conducted a non-parametric test, specifically the Friedman test, to assess the differences between the results. This test uses the ranks provided in Table 2. The test yielded a χ^2 score of 39.43 and an F_F score of 20.12. The critical value ($\alpha = 0.01$) ranged between 2.13 and 2.18 for 3 and 105 degrees of freedom, respectively. Given that both the chi-square and F_F scores were significantly higher than the critical value, we reject the null hypothesis. Consequently, we conclude that the models differ significantly from each other with a confidence level of 99%.

Subsequently, we employed the Nemenyi test [39] to pinpoint which pairs of classifiers exhibited significant differences. The computed critical difference was 0.656. The differences in the average ranks between D-ReLU and the other techniques are reported as follows: 0.86 for adversarial training, 1.89 for TRADES-1, and 1.14 for TRADES-6. Each of these differences surpasses the critical difference. Therefore, we conclude that classifiers utilizing D-ReLU demonstrate significantly greater robustness compared to those using all other methods.

6.3. Experimental Results for CIFAR100

Figure 14 illustrates the accuracy of various CNN architectures on clean CIFAR100 samples and adversarial examples generated by different white-box attacks. The figures reveal several important trends. Across all models, we observe a general pattern where the accuracy on clean samples remains relatively stable or slightly decreases as the balancer value increases. This stability indicates that the addition of the D-ReLU layer does not significantly compromise the model's performance on clean data, which is crucial for maintaining the overall utility of the model in non-adversarial settings.

There is a notable improvement in robustness with increasing balancer values for adversarial examples. This trend is consistent across all considered types of white-box attacks: FGSM, PGD, APGD_CE, APGD_DLR, and CW_L2. The accuracy on adversarial examples shows a significant upward trajectory, especially for higher balancer values, suggesting that the D-ReLU function effectively mitigates the impact of adversarial perturbations. This improvement in robustness is particularly pronounced in more complex models like ResNet50, ResNet101, MobilenetV2, and InceptionV3.



(**a**) Dense (**b**) Shallow CNN

(**c**) ResNet50 (**d**) ResNet101

(**e**) MobilenetV2 (**f**) InceptionV3

Figure 14. Accuracy of several types of networks on clean CIFAR100 and adversarial examples when adding a dense layer with a D-ReLU function before the output layer.

Table 3 shows a comparison between our approach and the other baselines concerning performance and robustness. Although the baselines outperform our approach in three architectures, our approach can provide more robust models than the other baselines in every case. Particularly in the cases of MobilenetV2 and InceptionV3, our approach exhibits notably superior performance compared to the other baselines.

Table 3. Accuracy metrics for multiple types of networks under various robust training schemes, evaluating them on both clean samples and adversarial examples generated by different adversarial attacks on the CIFAR100 dataset. Note that AP_{CE} is $APGD_{CE}$, AP_{DLR} is $APGD_{DLR}$, the accuracy metrics in bold are the highest in a specific model among the different training methods, the numbers in parentheses are the ranks for training methods under an architecture, TRADES-*k* indicates the TRADES approach with $\beta = k$, and D-ReLU-*k* represents the D-ReLU approach with $m = k$.

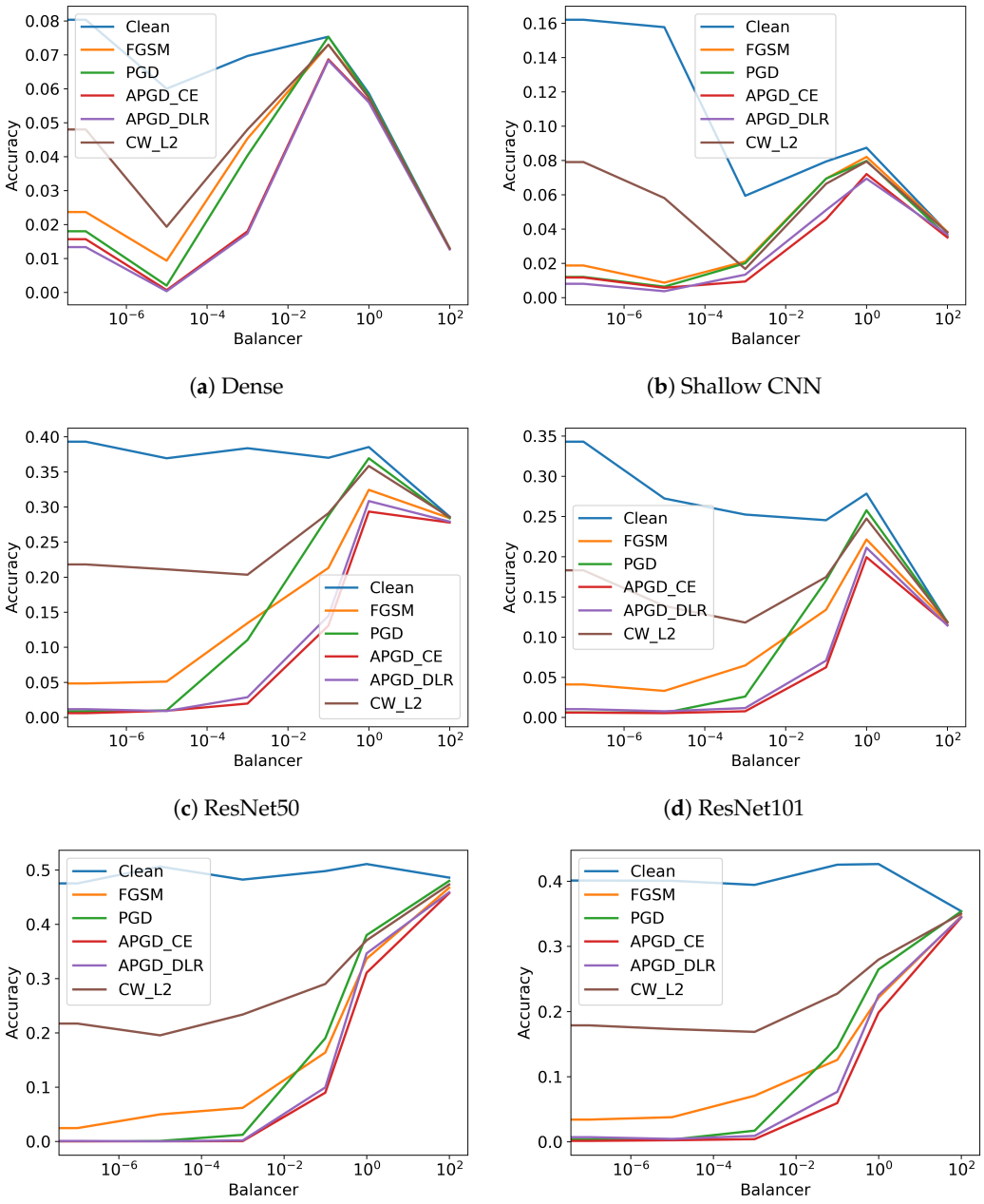
Model	Training	Clean %	FGSM %	PGD %	AP_{CE} %	AP_{DLR} %	CW_{L2} %
Dense	AT	**24.47** (1)	14.80 (2)	14.30 (2)	14.20 (2)	12.63 (2)	17.53 (2)
	TRADES-1	22.97 (3)	13.37 (4)	13.23 (4)	13.17 (4)	11.30 (4)	16.27 (4)
	TRADES-6	23.27 (2)	13.87 (3)	13.73 (3)	13.60 (3)	12.27 (3)	16.60 (3)
	D-ReLU-10^{-1}	21.47 (4)	**21.03** (1)	**21.00** (1)	**20.03** (1)	**19.77** (1)	**20.73** (1)
Shallow CNN	AT	**37.03** (1)	17.73 (3)	16.47 (3)	16.30 (3)	14.43 (3)	22.30 (3)
	TRADES-1	32.60 (3)	12.87 (4)	11.50 (4)	11.43 (4)	9.47 (4)	18.50 (4)
	TRADES-6	34.80 (2)	18.67 (2)	17.87 (2)	17.87 (2)	15.40 (2)	22.33 (2)
	D-ReLU-1	28.63 (4)	**27.53** (1)	**27.33** (1)	**24.87** (1)	**24.60** (1)	**27.23** (1)
ResNet50	AT	48.67 (3)	26.67 (3)	21.83 (3)	21.53 (3)	23.13 (3)	31.90 (2)
	TRADES-1	48.97 (2)	26.57 (4)	19.80 (4)	19.27 (4)	20.03 (4)	30.50 (4)
	TRADES-6	43.97 (4)	28.90 (2)	26.03 (2)	25.70 (2)	24.03 (2)	30.63 (3)
	D-ReLU-10^{2}	**52.33** (1)	**51.53** (1)	**52.47** (1)	**50.20** (1)	**51.17** (1)	**51.63** (1)
ResNet101	AT	44.97 (3)	23.57 (4)	18.67 (3)	18.33 (3)	18.77 (3)	27.77 (4)
	TRADES-1	**48.10** (1)	24.17 (3)	17.70 (4)	16.87 (4)	17.80 (4)	28.10 (3)
	TRADES-6	45.20 (2)	28.21 (2)	20.53 (2)	19.32 (2)	19.44 (2)	30.02 (2)
	D-ReLU-1	44.20 (4)	**39.03** (1)	**43.10** (1)	**37.33** (1)	**36.60** (1)	**40.63** (1)
MobilenetV2	AT	51.37 (2)	23.83 (3)	15.30 (3)	14.43 (3)	15.73 (2)	28.50 (2)
	TRADES-1	42.97 (3)	19.50 (4)	9.47 (4)	8.20 (4)	8.60 (4)	20.70 (4)
	TRADES-6	40.13 (4)	24.50 (2)	20.73 (2)	20.13 (2)	18.87 (3)	25.40 (3)
	D-ReLU-1	**56.40** (1)	**54.90** (1)	**55.07** (1)	**53.80** (1)	**54.17** (1)	**54.97** (1)
InceptionV3	AT	56.37 (3)	32.57 (4)	27.20 (3)	26.60 (3)	28.80 (3)	34.33 (4)
	TRADES-1	60.63 (2)	35.63 (2)	26.80 (4)	25.83 (4)	26.50 (4)	35.07 (2)
	TRADES-6	51.10 (4)	34.43 (3)	31.20 (2)	30.90 (2)	29.50 (2)	34.47 (3)
	D-ReLU-10^{2}	**67.07** (1)	**65.10** (1)	**64.43** (1)	**63.47** (1)	**63.70** (1)	**65.27** (1)

6.4. Experimental Results for TinyImagenet

Figure 15 presents the accuracy of several neural network architectures on clean Tiny-Imagenet samples and adversarial examples produced by various white-box attacks. The assessed networks include Dense and Shallow CNN, ResNet50, ResNet101, MobilenetV2, and InceptionV3. The experiments involved integrating a dense layer with a D-ReLU function before the output layer and varying the balancer value to observe its impact on model performance and robustness.

The graphs demonstrate a consistent pattern across all models, indicating the efficacy of the D-ReLU layer in enhancing adversarial robustness. On clean TinyImagenet samples, the accuracy generally remains stable or exhibits minor fluctuations as the balancer value changes. This stability suggests that the addition of the D-ReLU layer does not significantly impair the model's ability to correctly classify clean samples, maintaining its utility in standard scenarios.

For adversarial examples generated by white-box attacks (FGSM, PGD, APGD_CE, APGD_DLR, and CW_L2), there is a clear trend of improved robustness with increasing balancer values. The accuracy on these adversarial examples improves markedly, especially at higher balancer values, indicating that the D-ReLU function effectively counteracts the adversarial perturbations. This improvement is particularly evident in complex models like ResNet50, ResNet101, MobilenetV2, and InceptionV3, which show substantial gains in accuracy against adversarial attacks.



(**a**) Dense (**b**) Shallow CNN

(**c**) ResNet50 (**d**) ResNet101

(**e**) MobilenetV2 (**f**) InceptionV3

Figure 15. Accuracy of several types of networks on clean TinyImagenet and adversarial examples when adding a dense layer with a D-ReLU function before the output layer.

Table 4 shows the performance and robustness of our approach and the other baselines on the TinyImagenet dataset. The table also shows the ranking of the approaches in each architecture. Our approach struggles to find a balance between performance and robustness. However, in MobilenetV2, our approach outperforms the other ones in terms of performance and robustness.

Table 4. Accuracy metrics for multiple types of networks under various robust training schemes, evaluating them on both clean samples and adversarial examples generated by different adversarial attacks on the TinyImagenet dataset. Note that AP_{CE} is $APGD_{CE}$, AP_{DLR} is $APGD_{DLR}$, the accuracy metrics in bold are the highest in a specific model among the different training methods, the numbers in parentheses are the ranks for training methods under an architecture, TRADES-*k* indicates the TRADES approach with $\beta = k$, and D-ReLU-*k* represents the D-ReLU approach with $m = k$.

Model	Training	Clean %	FGSM %	PGD %	AP_{CE} %	AP_{DLR} %	CW_{L2} %
Dense	AT	8.63 (2)	5.40 (2)	5.13 (3)	5.00 (3)	4.27 (2)	7.00 (4)
	TRADES-1	8.57 (3)	4.80 (4)	4.77 (4)	4.73 (4)	4.10 (3)	**7.47** (1)
	TRADES-6	**8.70** (1)	5.07 (3)	5.13 (2)	5.10 (2)	3.93 (4)	7.30 (2)
	D-ReLU-10^{-1}	7.53 (4)	**7.30** (1)	**7.53** (1)	**6.87** (1)	**6.83** (1)	7.30 (3)
Shallow CNN	AT	**18.33** (1)	4.80 (2)	4.17 (2)	4.10 (2)	2.73 (2)	10.60 (2)
	TRADES-1	14.93 (3)	2.17 (4)	1.60 (4)	1.60 (4)	0.97 (4)	8.10 (3)
	TRADES-6	16.37 (2)	4.57 (3)	4.07 (3)	3.97 (3)	2.67 (3)	**10.63** (1)
	D-ReLU-1	8.40 (4)	**8.20** (1)	**7.97** (1)	**7.20** (1)	**6.93** (1)	7.93 (4)
ResNet50	AT	40.67 (3)	17.57 (4)	13.17 (4)	12.93 (4)	14.03 (4)	30.87 (4)
	TRADES-1	**48.10** (1)	22.15 (3)	16.10 (3)	15.55 (3)	14.95 (3)	**36.35** (1)
	TRADES-6	40.97 (2)	23.93 (2)	21.87 (2)	21.57 (2)	19.77 (2)	31.57 (3)
	D-ReLU-1	38.53 (4)	**32.43** (1)	**36.93** (1)	**29.33** (1)	**30.83** (1)	35.83 (2)
ResNet101	AT	32.73 (3)	15.43 (4)	13.10 (4)	12.63 (4)	11.40 (4)	24.17 (4)
	TRADES-1	**47.57** (1)	20.50 (3)	15.07 (3)	14.57 (3)	14.43 (3)	**34.73** (1)
	TRADES-6	39.13 (2)	**22.30** (1)	20.37 (2)	**20.03** (1)	17.67 (2)	30.63 (2)
	D-ReLU-1	27.83 (4)	22.13 (2)	**25.77** (1)	19.93 (2)	**21.10** (1)	24.73 (3)
MobilenetV2	AT	50.00 (2)	23.13 (3)	16.73 (3)	16.30 (3)	16.97 (3)	**37.73** (1)
	TRADES-1	48.87 (3)	20.60 (4)	13.57 (4)	12.83 (4)	12.00 (4)	35.10 (3)
	TRADES-6	43.20 (4)	23.70 (2)	21.23 (2)	20.87 (2)	19.03 (2)	33.73 (4)
	D-ReLU-1	**51.10** (1)	**33.63** (1)	**38.00** (1)	**31.07** (1)	**34.63** (1)	37.03 (2)
InceptionV3	AT	39.07 (4)	18.67 (4)	14.63 (4)	14.57 (4)	15.20 (4)	27.90 (4)
	TRADES-1	**60.43** (1)	**32.53** (1)	23.37 (3)	22.67 (2)	24.13 (2)	**46.17** (1)
	TRADES-6	50.43 (2)	32.03 (2)	**29.23** (1)	**28.90** (1)	**28.30** (1)	40.40 (2)
	D-ReLU-1	42.63 (3)	22.13 (3)	26.47 (2)	19.83 (3)	22.50 (3)	27.97 (3)

6.5. Discussion

The consistent improvements in adversarial robustness across the MNIST, CIFAR10, CIFAR100, and TinyImagenet datasets highlight several key implications.

First, the D-ReLU layer's effectiveness across different datasets and model architectures indicates its broad applicability. It suggests that this technique can be reliably used to enhance the adversarial robustness of various neural networks without specific tailoring to individual datasets.

Second, despite the significant gains in adversarial robustness, the performance on clean samples remains largely unaffected. This balance ensures that the models remain useful and reliable in standard conditions, which is critical for practical deployment.

Third, the approach scales well with model complexity. More advanced models like ResNet and InceptionV3, which are typically used in real-world applications, benefit greatly from the addition of a D-ReLU layer, showing substantial improvements in defending against sophisticated white-box attacks.

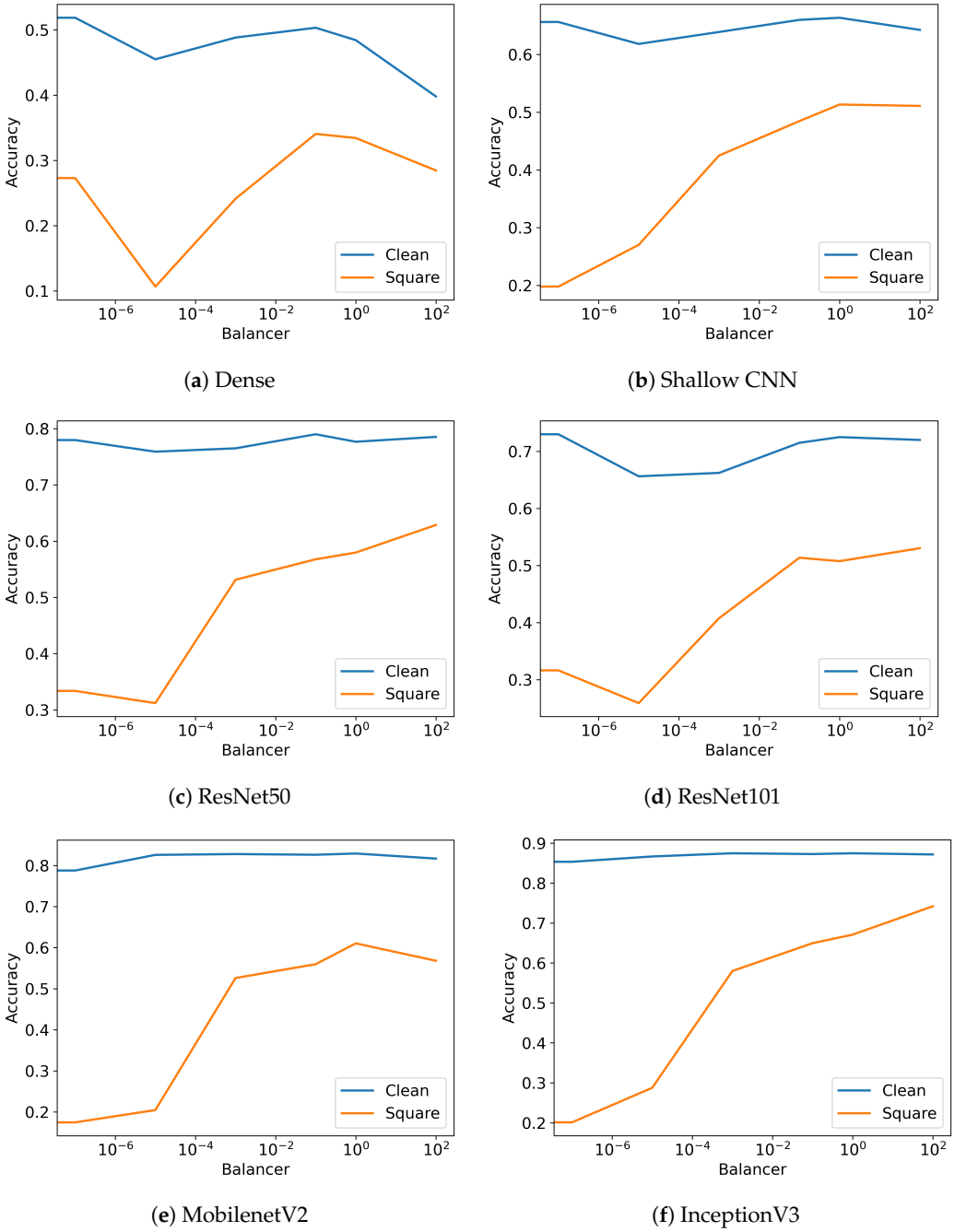
Moreover, by effectively countering a range of white-box attacks, the D-ReLU layer enhances the overall security of neural networks. This makes it a valuable addition to the suite of techniques aimed at protecting models against adversarial threats.

The integration of a dense layer with a D-ReLU function before the output layer provides a robust defense mechanism against white-box attacks across the MNIST, CIFAR10, CIFAR100, and TinyImagenet datasets. This approach ensures that neural networks can maintain high

performance on clean samples while significantly improving their resilience to adversarial perturbations, thereby enhancing their reliability and security in various applications.

7. Black-Box Attack Experiments

In addition to the promising results against white-box attacks, we also evaluated the performance of the D-ReLU function in enhancing the robustness of CNNs against black-box attacks, specifically the Square attack. Figures 16–18 offer valuable insights into how D-ReLU impacts various models across different datasets under black-box attack scenarios.

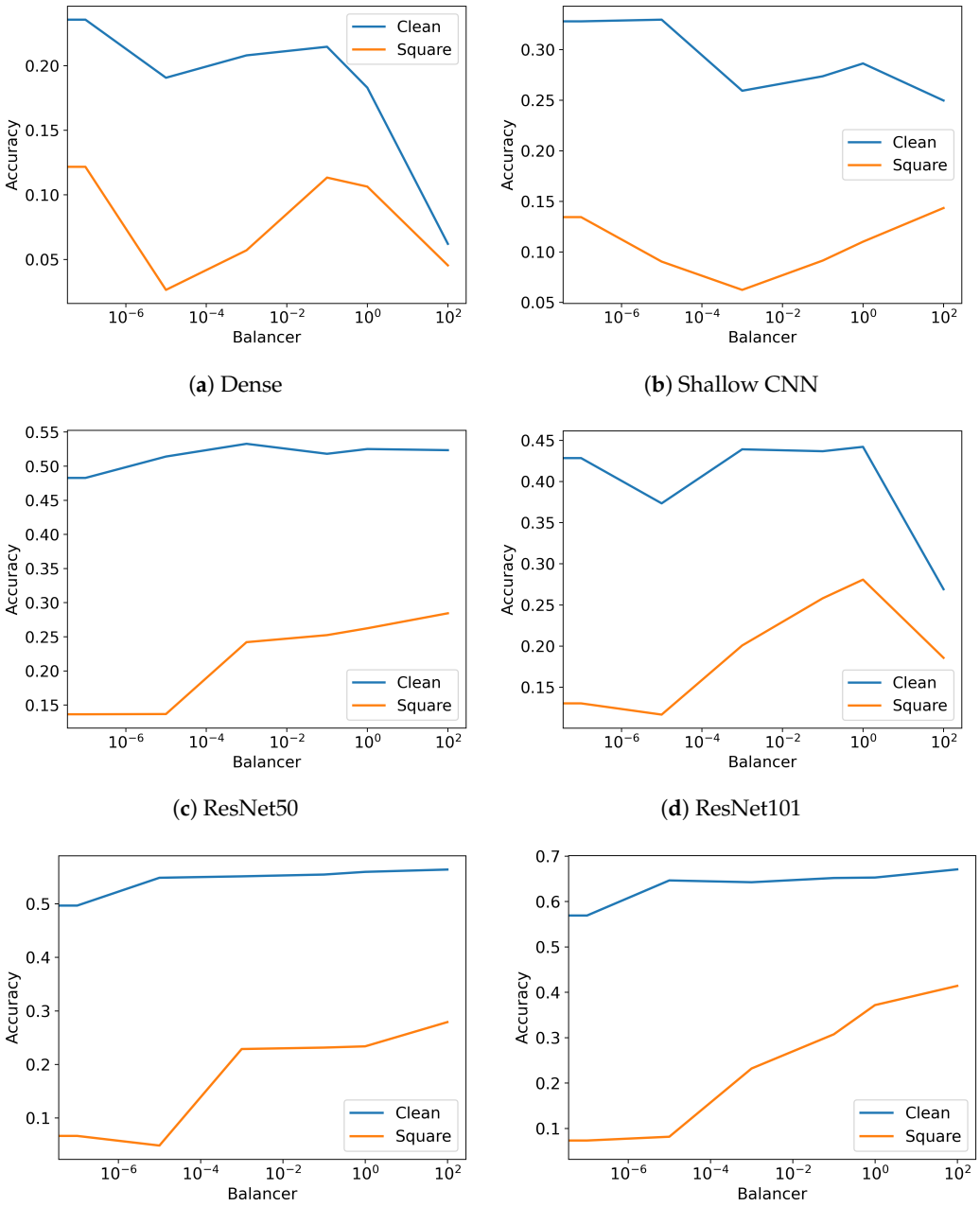


(**a**) Dense (**b**) Shallow CNN

(**c**) ResNet50 (**d**) ResNet101

(**e**) MobilenetV2 (**f**) InceptionV3

Figure 16. Accuracy of several types of networks on clean CIFAR10 and adversarial examples generated by a black-box attack (i.e., square attack) when adding a dense layer with a D-ReLU function before the output layer.

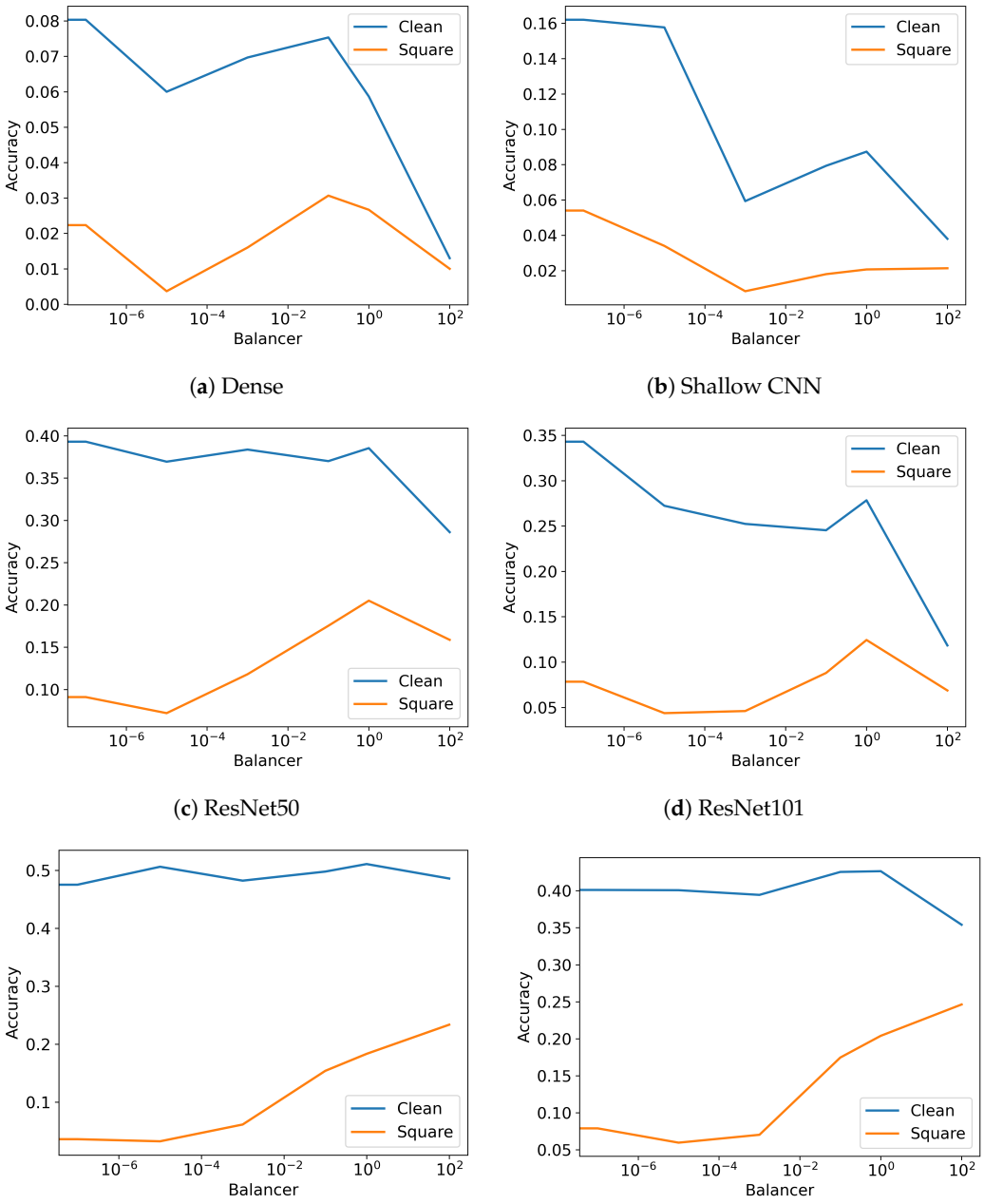


(**a**) Dense (**b**) Shallow CNN

(**c**) ResNet50 (**d**) ResNet101

(**e**) MobilenetV2 (**f**) InceptionV3

Figure 17. Accuracy of several types of networks on clean CIFAR100 and adversarial examples generated by a black-box attack (i.e., square attack) when adding a dense layer with a D-ReLU function before the output layer.



(**a**) Dense (**b**) Shallow CNN

(**c**) ResNet50 (**d**) ResNet101

(**e**) MobilenetV2 (**f**) InceptionV3

Figure 18. Accuracy of several types of networks on clean TinyImagenet and adversarial examples generated by a black-box attack (i.e., square attack) when adding a dense layer with a D-ReLU function before the output layer.

7.1. Experimental Results for CIFAR10

In Figure 16, the accuracy of several network types on clean CIFAR10 data and adversarial examples generated by the black-box attack is depicted. For dense networks (Figure 16a), the accuracy on clean samples remains relatively stable across different balancer values. However, the accuracy against adversarial examples shows a notable improvement with increasing balancer values, indicating enhanced robustness. Shallow CNNs (Figure 16b) display a similar pattern, with a significant improvement in adversarial robustness at higher balancer values, while the clean accuracy remains consistent.

ResNet50 and ResNet101 (Figure 16c,d) both demonstrate substantial gains in adversarial robustness with increasing balancer values. This trend suggests that deeper networks benefit more from the D-ReLU layer in terms of adversarial resilience. MobilenetV2 (Figure 16e) also shows consistent improvement in adversarial accuracy with higher balancer values, despite slight fluctuations in clean accuracy. InceptionV3 (Figure 16f) exhibits a strong increase in adversarial robustness with higher balancer values while maintaining high accuracy on clean samples.

7.2. Experimental Results for CIFAR100

Figure 17 presents the accuracy metrics for CIFAR100. Dense networks (Figure 17a) show moderate improvement in adversarial robustness with the addition of the D-ReLU layer, though clean accuracy remains largely unaffected. Shallow CNNs (Figure 17b) follow a clear trend of increasing adversarial accuracy with higher balancer values, indicating the D-ReLU layer's effectiveness in enhancing robustness.

For deeper networks like ResNet50 and ResNet101 (Figures 16c and 17d), there is improved adversarial robustness with increasing balancer values, though a slight decrease in clean accuracy is observed at higher balancer values. MobilenetV2 (Figure 17e) displays marked improvement in adversarial robustness with higher balancer values, with minimal fluctuations in clean accuracy. InceptionV3 (Figure 17f) shows the highest gains in adversarial robustness, maintaining strong performance on clean samples.

7.3. Experimental Results for TinyImagenet

In Figure 18, the results for TinyImagenet are detailed. Dense networks (Figure 18a) show a significant increase in adversarial robustness with higher balancer values, while clean accuracy remains stable. Shallow CNNs (Figure 18b) exhibit improved adversarial accuracy with higher balancer values, though clean accuracy shows some variability.

Deeper networks like ResNet50 and ResNet101 (Figure 18c,d) benefit significantly in terms of adversarial robustness with increasing balancer values, with slight fluctuations in clean accuracy. MobilenetV2 (Figure 18e) demonstrates notable improvement in adversarial robustness with higher balancer values, with clean accuracy remaining relatively unaffected. InceptionV3 (Figure 18f) shows the most substantial gains in adversarial robustness among all tested architectures, with clean accuracy remaining high.

7.4. Comparison to Other Baselines

Table 5 provides accuracy metrics and rankings for various neural network models trained under different robust training schemes and evaluated on clean samples, as well as adversarial examples generated by a black-box attack (denoted as Square), on the CIFAR10, CIFAR100, and TinyImagenet datasets. The displayed values are percentages, with the highest accuracy metrics highlighted in bold for each specific model among the different training methods.

The TRADES-6 strategy demonstrates superior performance across most scenarios in the dense network. In the Shallow CNN architecture, the D-ReLU method showcases a competitive edge over TRADES-based approaches specifically on the CIFAR10 dataset. However, TRADES-6 surpasses D-ReLU in other instances. For the ResNet50, MobilenetV2, and InceptionV3 models, D-ReLU stands out as the top performer on the CIFAR10 and CIFAR100 datasets. Nevertheless, its efficiency on the TinyImagenet dataset falls short in comparison to the TRADES-based techniques, highlighting a trade-off between performance and robustness. ResNet101 presents a mix of results, showcasing variability in its performance outcomes.

Table 5. Accuracy metrics for multiple types of networks under various robust training schemes, evaluated on both clean samples and adversarial examples generated by a black-box attach (i.e., Square) on the CIFAR10, CIFAR100, and TinyImagenet datasets. Note that the accuracy metrics in bold are the highest in a specific model among the different training methods, the numbers in parentheses are the ranks for training methods under an architecture, and TRADES-k indicates the TRADES approach with $\beta = k$.

Model	Training	CIFAR10		CIFAR100		TinyImagenet	
		Clean %	Square %	Clean %	Square %	Clean %	Square %
Dense	TRADES-1	**52.33** (1)	34.03 (2)	22.97 (2)	13.90 (2)	8.57 (2)	4.80 (2)
	TRADES-6	51.30 (2)	**38.47** (1)	**23.27** (1)	**14.13** (1)	**8.70** (1)	**4.87** (1)
	D-ReLU	48.43 (3)	33.43 (3)	21.47 (3)	11.33 (3)	7.53 (3)	3.07 (3)
Shallow CNN	TRADES-1	**67.37** (1)	45.93 (3)	32.60 (3)	15.47 (2)	14.93 (3)	5.43 (2)
	TRADES-6	64.50 (3)	49.30 (2)	**34.80** (1)	**19.70** (1)	**16.37** (1)	**7.13** (1)
	D-ReLU	66.37 (2)	**51.33** (1)	32.87 (2)	13.53 (3)	16.20 (2)	5.40 (3)
ResNet50	TRADES-1	75.70 (2)	50.70 (3)	48.97 (2)	25.10 (3)	**48.40** (1)	**26.53** (1)
	TRADES-6	71.63 (3)	53.57 (2)	43.97 (3)	27.03 (2)	40.97 (2)	25.03 (2)
	D-ReLU	**78.53** (1)	**62.87** (1)	**52.33** (1)	**28.43** (1)	38.53 (3)	20.50 (3)
ResNet101	TRADES-1	**74.60** (1)	45.37 (2)	**48.10** (1)	23.20 (2)	**47.57** (1)	**25.07** (1)
	TRADES-6	66.67 (3)	43.63 (3)	10.67 (3)	1.67 (3)	39.13 (2)	24.00 (2)
	D-ReLU	72.00 (2)	**53.03** (1)	44.20 (2)	**28.07** (1)	27.83 (3)	12.43 (3)
MobilenetV2	TRADES-1	73.13 (2)	43.13 (3)	42.97 (2)	15.40 (3)	48.87 (2)	25.00 (2)
	TRADES-6	68.60 (3)	49.17 (2)	40.13 (3)	22.30 (2)	43.20 (3)	**26.23** (1)
	D-ReLU	**82.90** (1)	**61.03** (1)	**56.40** (1)	**27.90** (1)	**51.10** (1)	18.33 (3)
InceptionV3	TRADES-1	82.53 (2)	64.17 (2)	60.63 (2)	34.50 (2)	**60.43** (1)	**39.60** (1)
	TRADES-6	76.97 (3)	62.40 (3)	51.10 (3)	34.03 (3)	50.43 (2)	36.10 (2)
	D-ReLU	**87.17** (1)	**74.20** (1)	**67.07** (1)	**41.40** (1)	42.63 (3)	24.63 (3)

7.5. Discussion

The effectiveness of D-ReLU against black-box attacks has several important implications. First, it highlights the potential of D-ReLU to provide robust defenses in more realistic adversarial settings where attackers lack full knowledge of the model's parameters and architecture. This makes D-ReLU a valuable tool for real-world applications where security and reliability are paramount.

Second, the consistent improvement in robustness across different architectures and datasets suggests that D-ReLU can be widely applied to various deep learning models, making it a versatile and scalable solution for enhancing adversarial defenses.

Lastly, the ability of D-ReLU to improve robustness without compromising performance on clean samples is particularly noteworthy, especially on the CIFAR10 and CIFAR100 datasets. This balance between robustness and accuracy ensures that models remain effective for their intended tasks while being resilient to adversarial perturbations. However, it is still difficult to train the model with D-ReLU on a large dataset like the TinyImagenet dataset.

Overall, the findings underscore the robustness of the D-ReLU function against black-box attacks, further validating its utility in strengthening the security of deep learning models in diverse and practical scenarios. This reinforces the importance of integrating such robust functions into model architectures to safeguard against a wide range of adversarial threats.

8. Experiments with Augmented Dataset

The study conducted by Wang et al. [40], as highlighted within the extensive literature review, has brought to light the significant impact of incorporating the elucidating diffusion model (EDM) proposed by Karras et al. [41] as a means to effectively mitigate the prevalent issue of overfitting encountered during adversarial training processes. By augmenting the training dataset with the EDM, promising results have been observed in terms of enhancing the robustness and generalization capabilities of the learning model. Against this backdrop, the subsequent analysis presented in this section undertakes a comprehensive evaluation through comparative studies between our proposed methodology and the renowned TRADES technique introduced by Zhang et al. [36]. This comparative analysis is conducted utilizing the augmented training samples, demonstrating the efficacy and superiority of our approach in bolstering the resilience of the learning system against adversarial attacks and enhancing overall performance metrics.

In every epoch, a combination of generated samples and original training samples is utilized. As outlined in the research conducted by Wang et al. [40], a specific configuration is followed for the CIFAR10 and CIFAR100 datasets. Here, a random selection process is employed to choose samples from both the original dataset and the generated samples. Approximately 30% of the training samples are sourced from the original dataset, while the remaining samples are from the generated dataset. It is imperative to note that despite this mixing process, the overall size of the training dataset remains constant.

Furthermore, the research also stipulates the use of a hyperparameter value of $\beta = 5$ for the TRADES method. Moving on to the TinyImagenet dataset, a slightly different approach is adopted. In this case, 20% of the training samples are sourced from the original dataset, with the remaining samples coming from the generated dataset. Consistent with the literature by Wang et al. (2023) [40], a value of $\beta = 8$ is utilized for the TRADES method in this context. To ensure a fair comparison, the same $\beta = 5$ value is also utilized in this scenario.

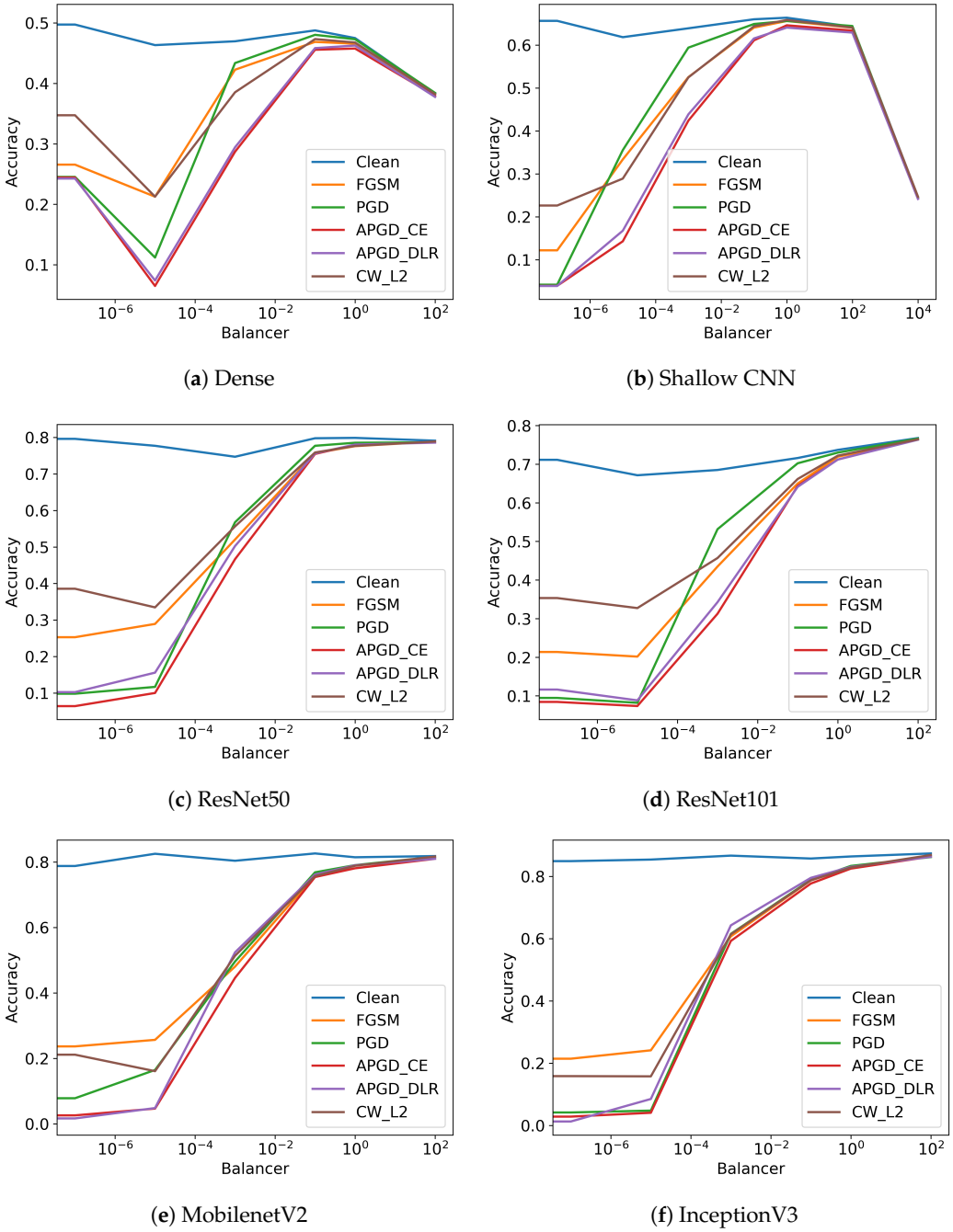
8.1. Experimental Results

The visual representations displayed in Figure 19 for CIFAR10 and Figure 20 for CIFAR100 offer an insightful analysis of the performance and robustness of various architectures trained with D-ReLU under white-box attacks, leveraging a training dataset enriched with generated samples from the EDM. The fusion of D-ReLU with the EDM showcases impressive results on both the CIFAR10 and CIFAR100 datasets, particularly demonstrating significant efficacy when applied to deep architectures. Notably, the combined approach of D-ReLU plus EDM exhibits remarkable performance and robustness; especially noteworthy is how it outperforms instances where D-ReLU is employed without the integration of the EDM.

Intriguingly, even at higher values of m, such as $m = 100$, the performance and robustness metrics do not exhibit a notable decline as observed with the utilization of D-ReLU in isolation, underscoring the added value and efficacy of incorporating EDM-generated samples into the training set. This observation highlights the positive impact of integrating EDM in the training process, particularly in enhancing the overall performance and robustness of deep architectures across the CIFAR10 and CIFAR100 datasets. Such findings provide valuable insights into the effectiveness of synergistic methods like D-ReLU plus EDM in improving the learning capabilities and resilience of neural network models.

Tables 6 and 7 provide a comparative analysis between our approach using D-ReLU and the TRADES method with generated samples from the EDM across the CIFAR10 and CIFAR100 datasets, respectively. The tables also show the rankings for comparison. They also provide a comparative analysis of our approach using D-ReLU with the TRADES method with generated samples from the EDM across the CIFAR10 and CIFAR100 datasets. When considering the CIFAR10 dataset, it is evident that D-ReLU generally surpasses TRADES regarding the robustness of the models in a majority of the scenarios. The exception lies in cases involving smaller network architectures such as Dense and Shallow

CNNs, where TRADES demonstrates noticeably superior performance compared to D-ReLU. In contrast, D-ReLU shows its strengths in deeper network architectures, where its performance is on par with or even exceeds that of TRADE. This trend of comparative performance is not isolated to the CIFAR10 dataset but is also observable in the results for the CIFAR100 dataset.

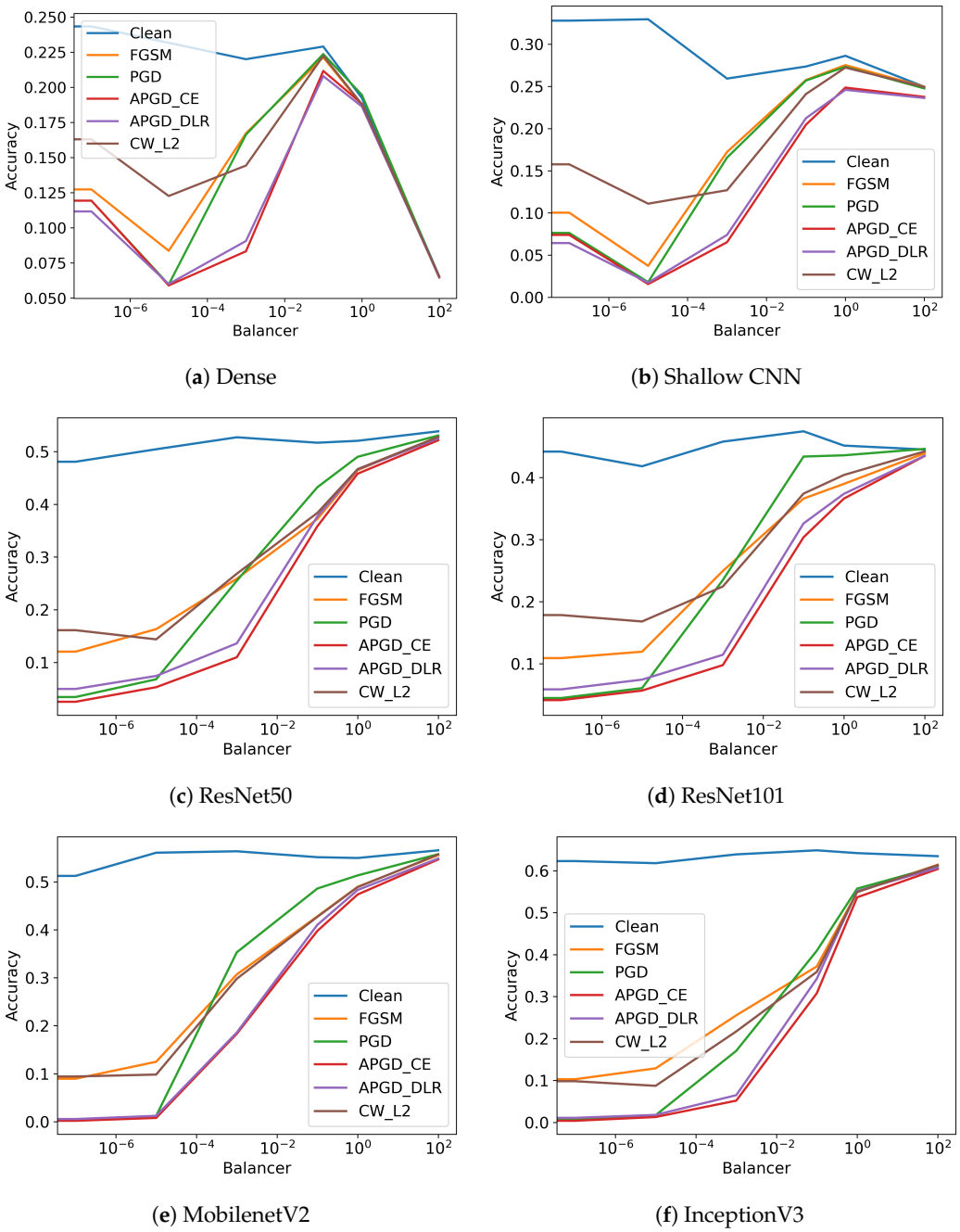


(**a**) Dense (**b**) Shallow CNN

(**c**) ResNet50 (**d**) ResNet101

(**e**) MobilenetV2 (**f**) InceptionV3

Figure 19. Accuracy of several types of networks on clean CIFAR10 and adversarial examples when adding a dense layer with a D-ReLU function before the output layer and training them with augmented data samples generated from the EDM.



(**a**) Dense (**b**) Shallow CNN

(**c**) ResNet50 (**d**) ResNet101

(**e**) MobilenetV2 (**f**) InceptionV3

Figure 20. Accuracy of several types of networks on clean CIFAR100 and adversarial examples when adding a dense layer with a D-ReLU function before the output layer and training them with augmented data samples generated from the EDM.

For deeper evaluations, the performance differential between D-ReLU and TRADES across different network depths highlights the significance of choosing appropriate defensive techniques depending on the complexity and depth of the employed models. Further insights suggest that while TRADES tends to be more effective with simpler, less deep networks, D-ReLU offers competitive advantages, primarily in more complex architectures. This pattern suggests that the underlying mechanisms of D-ReLU might be better tuned for managing the higher complexities and intricacies associated with deeper networks. Hence, assessing the networks' architecture becomes crucial when implementing robust training methods, as the choice between D-ReLU and TRADES could significantly impact the effectiveness of model robustness against adversarial attacks.

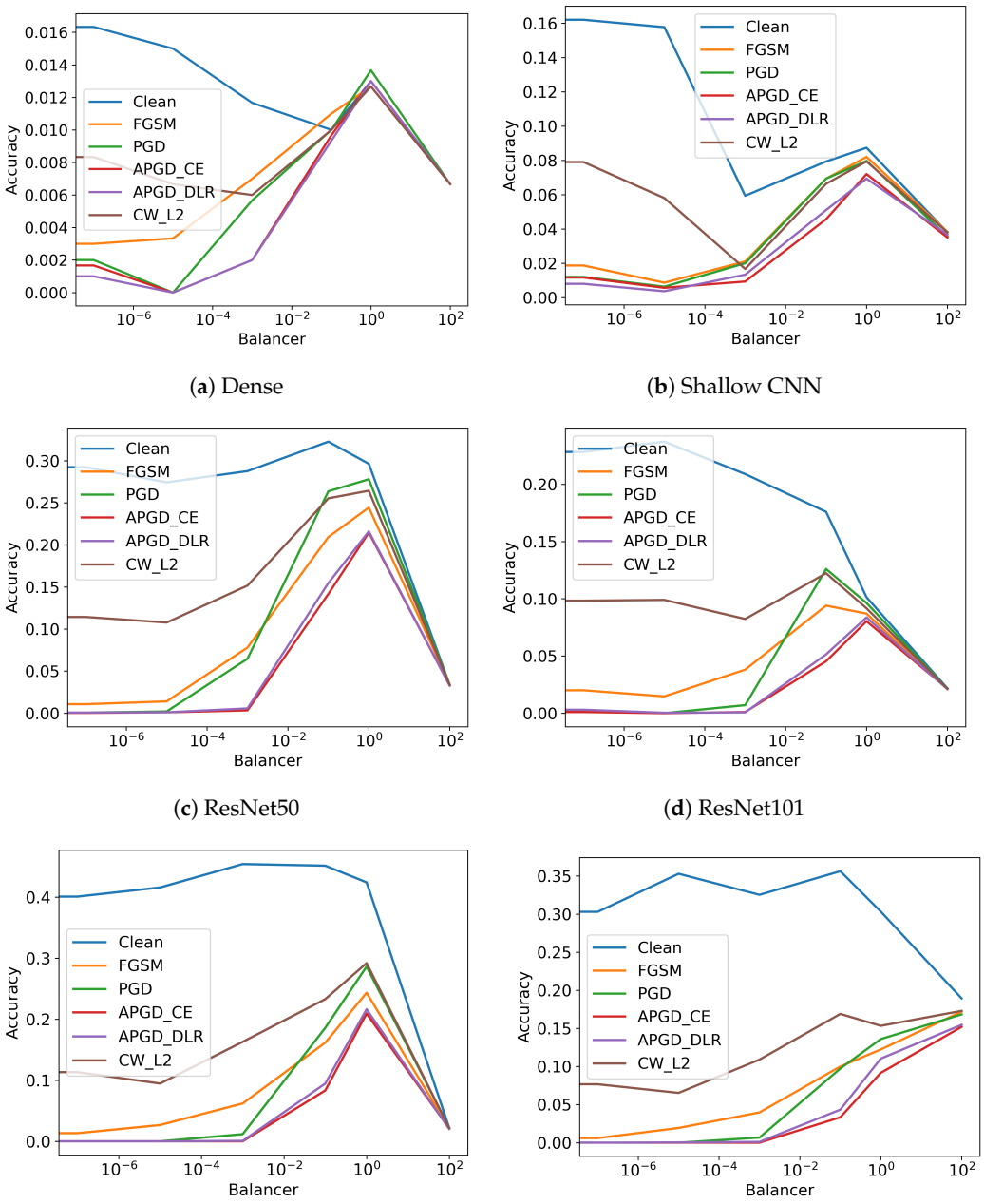
Table 6. Accuracy metrics for multiple types of networks under various robust training schemes with generated samples from the EDM, evaluating them on both clean samples and adversarial examples generated by different white-box attacks on the CIFAR10 dataset. Note that the accuracy metrics in bold are the highest in a specific model among the different training methods, and the numbers in parentheses are the ranks for training methods under an architecture.

Model	Training	Clean %	FGSM %	PGD %	$APGD_{CE}$ %	$APGD_{DLR}$ %	CW_{L2} %
Dense	D-ReLU	48.47 (2)	**46.87** (1)	**48.03** (1)	45.57 (2)	**45.83** (1)	47.33 (2)
	TRADES	**62.47** (1)	46.67 (2)	46.07 (2)	**46.13** (1)	44.63 (2)	**52.8** (1)
Shallow CNN	D-ReLU	67.97 (2)	**66.57** (1)	**67.07** (1)	**65.4** (1)	**65.4** (1)	**66.97** (1)
	TRADES	**74.3** (1)	59.03 (2)	57.93 (2)	57.93 (2)	56.53 (2)	63.6 (2)
ResNet50	D-ReLU	79.1 (2)	**78.87** (1)	**78.67** (1)	**78.63** (1)	**78.57** (1)	**78.87** (1)
	TRADES	**80.6** (1)	66.77 (2)	65.97 (2)	65.5 (2)	64.03 (2)	70.2 (2)
ResNet101	D-ReLU	76.77 (2)	**76.37** (1)	**76.63** (1)	**76.43** (1)	**76.33** (1)	**76.43** (1)
	TRADES	**77.97** (1)	63.43 (2)	61.93 (2)	61.87 (2)	59.77 (2)	67.33 (2)
MobilenetV2	D-ReLU	**81.8** (1)	**81.47** (1)	**81.6** (1)	**80.97** (1)	**80.97** (1)	**81.67** (1)
	TRADES	79.33 (2)	62.27 (2)	61.1 (2)	60.67 (2)	58.4 (2)	66.87 (2)
InceptionV3	D-ReLU	87.4 (2)	**86.77** (1)	**86.23** (1)	**86.4** (1)	**86.33** (1)	**86.9** (1)
	TRADES	**87.73** (1)	74.53 (2)	73.17 (2)	73.07 (2)	72.1 (2)	75.93 (2)

Table 7. Accuracy metrics for multiple types of networks under various robust training schemes with generated samples from the EDM, evaluating them on both clean samples and adversarial examples generated by different white-box attacks on the CIFAR100 dataset. Note that the accuracy metrics in bold are the highest in a specific model among the different training methods, and the numbers in parentheses are the ranks for training methods under an architecture.

Model	Training	Clean %	FGSM %	PGD %	$APGD_{CE}$ %	$APGD_{DLR}$ %	CW_{L2} %
Dense	D-ReLU	22.90 (2)	22.13 (2)	22.37 (2)	21.17 (2)	20.80 (2)	22.23 (2)
	TRADES	**36.03** (1)	**23.93** (1)	**23.57** (1)	**23.47** (1)	**22.13** (1)	**26.97** (1)
Shallow CNN	D-ReLU	32.20 (2)	**31.50** (1)	**31.70** (1)	28.57 (2)	**28.50** (1)	31.03 (2)
	TRADES	**44.23** (1)	29.90 (2)	29.33 (2)	29.30 (1)	26.93 (2)	**33.90** (1)
ResNet50	D-ReLU	53.83 (2)	**52.8** (1)	**53.03** (1)	**52.13** (1)	**52.50** (1)	**52.77** (1)
	TRADES	**55.33** (1)	40.17 (2)	38.03 (2)	37.80 (2)	37.27 (2)	43.13 (2)
ResNet101	D-ReLU	44.50 (2)	**43.90** (1)	**44.60** (1)	**43.47** (1)	**43.50** (1)	**44.20** (1)
	TRADES	**52.60** (1)	37.73 (2)	36.23 (2)	36.03 (2)	34.57 (2)	41.27 (2)
MobilenetV2	D-ReLU	**56.57** (1)	**55.57** (1)	**55.77** (1)	**54.67** (1)	**54.87** (1)	**55.70** (1)
	TRADES	51.27 (2)	38.57 (2)	37.10 (2)	36.73 (2)	35.50 (2)	40.90 (2)
InceptionV3	D-ReLU	**63.47** (1)	**61.43** (1)	**61.07** (1)	**60.40** (1)	**60.70** (1)	**61.33** (1)
	TRADES	62.90 (2)	48.33 (2)	46.5 (2)	46.23 (2)	45.67 (2)	49.43 (2)

The graphical representation provided in Figure 21 presents a detailed evaluation of the outcomes derived from implementing D-ReLU in conjunction with the EDM on the TinyImagenet dataset. Interestingly, the results indicate noticeable discrepancies in both performance and robustness compared to scenarios where solely D-ReLU is deployed. This inferior performance observed in the approach combining D-ReLU with the EDM can be attributed to a crucial factor: the generated samples utilized for augmentation originate from data points that are external to the test dataset.



(**a**) Dense (**b**) Shallow CNN

(**c**) ResNet50 (**d**) ResNet101

(**e**) MobilenetV2 (**f**) InceptionV3

Figure 21. Accuracy of several types of networks on clean TinyImagenet and adversarial examples when adding a dense layer with a D-ReLU function before the output layer and training them with augmented data samples generated from the EDM.

The discrepancy in results between the D-ReLU with EDM method and the standalone D-ReLU approach on the TinyImagenet dataset underscores the significance of the source of generated samples in the training process. By incorporating samples that do not align closely with the original dataset, the model may encounter challenges in effectively generalizing and adapting to the unseen data during inference. This discrepancy highlights the critical aspect of data-source relevance in the augmentation process, emphasizing the importance of utilizing samples that are representative of the original dataset to ensure optimal performance and robustness in model training.

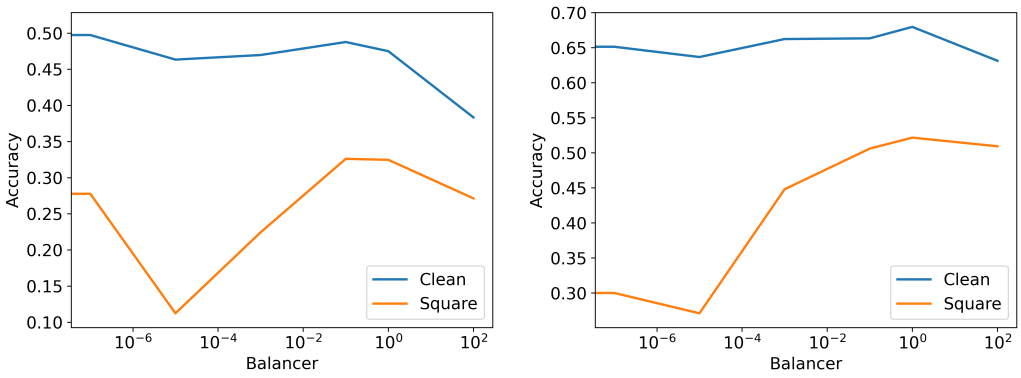
Table 8 presents a detailed comparison of the D-ReLU and TRADES training methodologies using samples generated from the EDM approach, particularly within the context of the TinyImagenet dataset. Upon examining the results, it becomes noticeable that the performance of D-ReLU in smaller network structures, such as Dense and Shallow CNNs, is substantially deficient. When employing D-ReLU in these compact network configurations, the results indicate a stark underperformance compared to its counterpart, TRADES, which appears to better handle the constraints and demands posed by smaller neural networks.

Table 8. Accuracy metrics for multiple types of networks under various robust training schemes with generated samples from the EDM, evaluating them on both clean samples and adversarial examples generated by different white-box attacks on the TinyImagenet dataset. Note that the accuracy metrics in bold are the highest in a specific model among the different training methods, and the numbers in parentheses are the ranks for training methods under an architecture.

Model	Training	Clean %	FGSM %	PGD %	$APGD_{CE}$ %	$APGD_{DLR}$ %	CW_{L2} %
Dense	D-ReLU	1.3 (2)	**1.27** (1)	**1.37** (1)	**1.3** (1)	**1.3** (1)	1.27 (2)
	TRADES	**2.4** (1)	1.07 (2)	1.07 (2)	1.03 (2)	0.8 (2)	**1.77** (1)
Shallow CNN	D-ReLU	1.87 (2)	1.77 (2)	1.77 (2)	1.5 (2)	**1.53** (1)	1.8 (2)
	TRADES	**7.33** (1)	**1.97** (1)	**1.87** (1)	**1.87** (1)	1.13 (2)	**4.6** (1)
ResNet50	D-ReLU	**29.63** (1)	**24.43** (1)	**27.8** (1)	**21.47** (1)	**21.6** (1)	**26.43** (1)
	TRADES	8.63 (2)	4.13 (2)	3.7 (2)	3.57 (2)	2.9 (2)	5.97 (2)
ResNet101	D-ReLU	**17.6** (1)	**9.4** (1)	**12.6** (1)	**4.53** (1)	**5.13** (1)	**12.23** (1)
	TRADES	7.3 (2)	3.63 (2)	3.37 (2)	3.33 (2)	2.87 (2)	5.2 (2)
MobilenetV2	D-ReLU	**42.43** (1)	**24.43** (1)	**28.63** (1)	**20.93** (1)	**21.63** (1)	**29.2** (1)
	TRADES	18.13 (2)	8 (2)	7.03 (2)	6.63 (2)	5.2 (2)	12.63 (2)
InceptionV3	D-ReLU	**35.63** (1)	**10** (1)	**9.73** (1)	3.33 (2)	**4.33** (1)	**16.9** (1)
	TRADES	12.2 (2)	5.57 (2)	5.07 (2)	**5** (1)	4.3 (2)	7.63 (2)

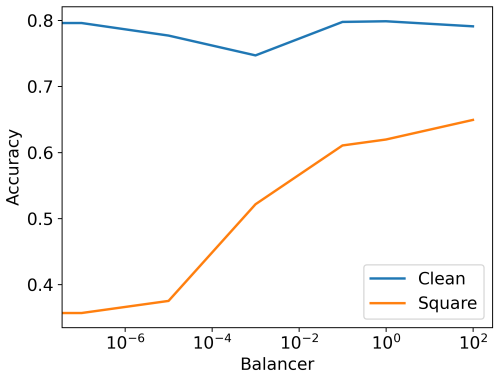
Conversely, in the context of more elaborate and deep network architectures, D-ReLU demonstrates a marked superiority, substantially outperforming TRADES. This significant enhancement in performance with deep networks suggests that D-ReLU is particularly well suited to leverage the complex structures and layers involved in such models, potentially exploiting deeper features and more intricate decision boundaries that deeper architectures facilitate.

Figures 22–24 visualize the accuracy on the clean and adversarial samples under several architectures on the CIFAR10, CIFAR100, and TinyImagenet datasets. These results follow the same patterns as in the white-box attacks.

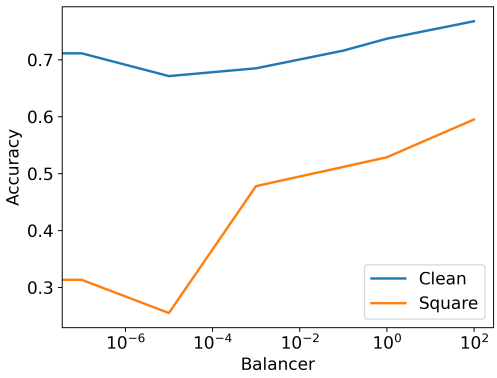


(**a**) Dense (**b**) Shallow CNN

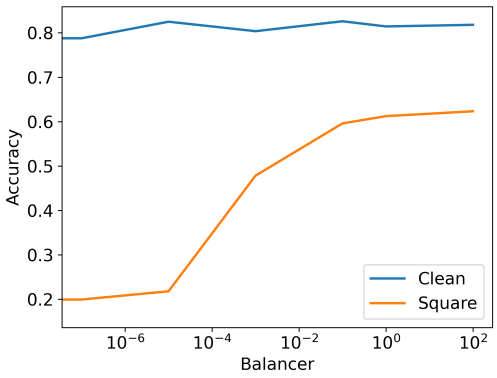
Figure 22. *Cont.*



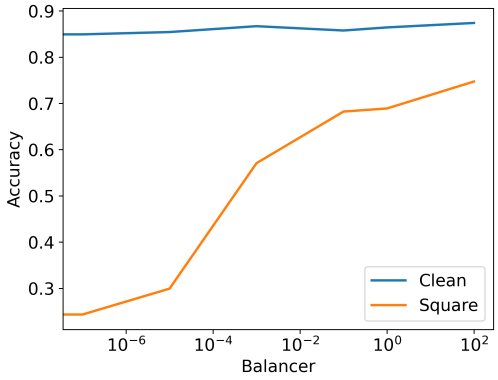
(**c**) ResNet50



(**d**) ResNet101

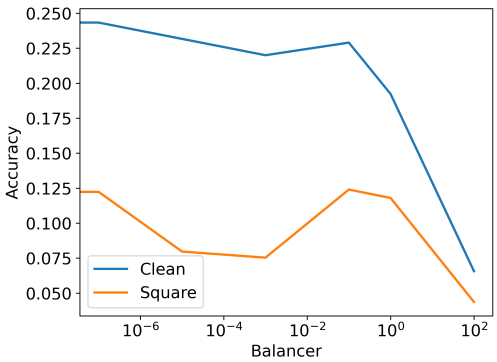


(**e**) MobilenetV2

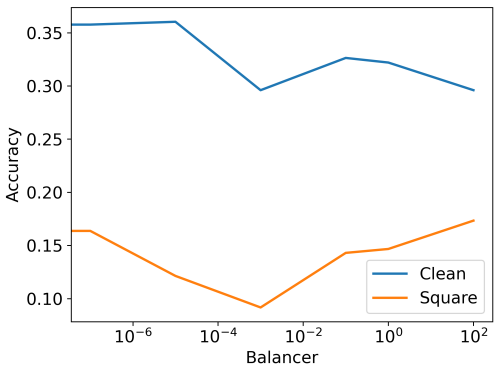


(**f**) InceptionV3

Figure 22. Accuracy of several types of networks on clean CIFAR10 and adversarial examples generated by a black-box attack (i.e., square attack) when adding a dense layer with a D-ReLU function before the output layer and training them with augmented data samples generated from the EDM.

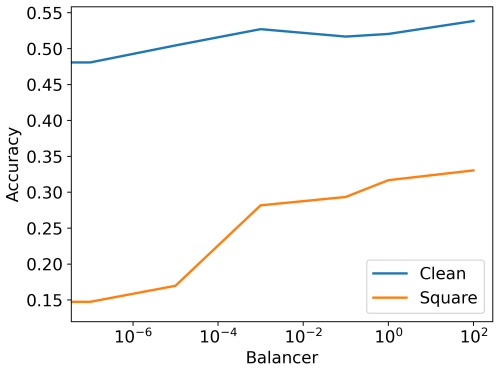


(**a**) Dense

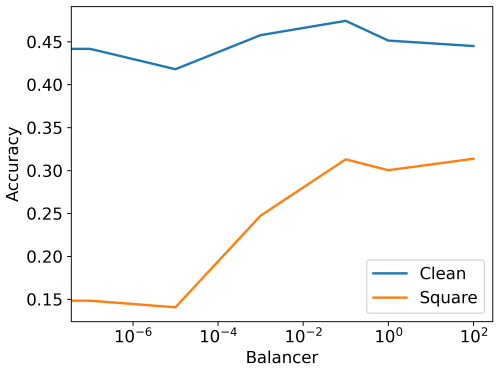


(**b**) Shallow CNN

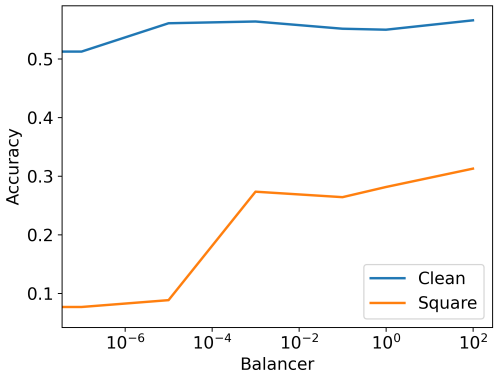
Figure 23. *Cont.*



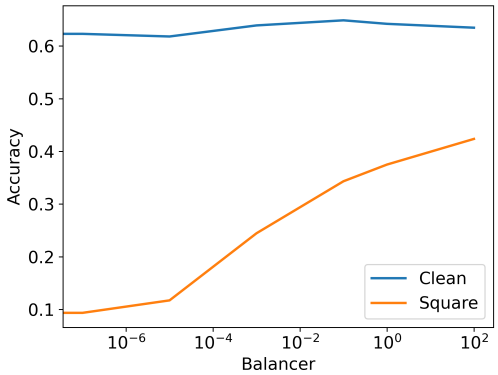
(**c**) ResNet50



(**d**) ResNet101

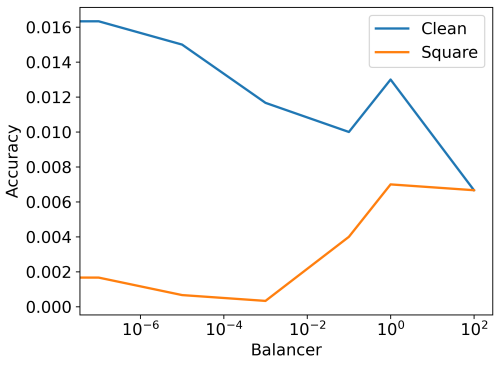


(**e**) MobilenetV2

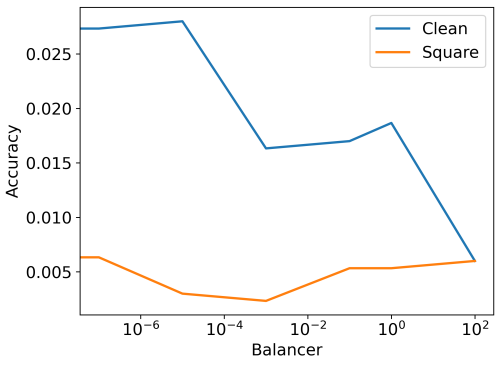


(**f**) InceptionV3

Figure 23. Accuracy of several types of networks on clean CIFAR100 and adversarial examples generated by a black-box attack (i.e., square attack) when adding a dense layer with a D-ReLU function before the output layer and training them with augmented data samples generated from the EDM.

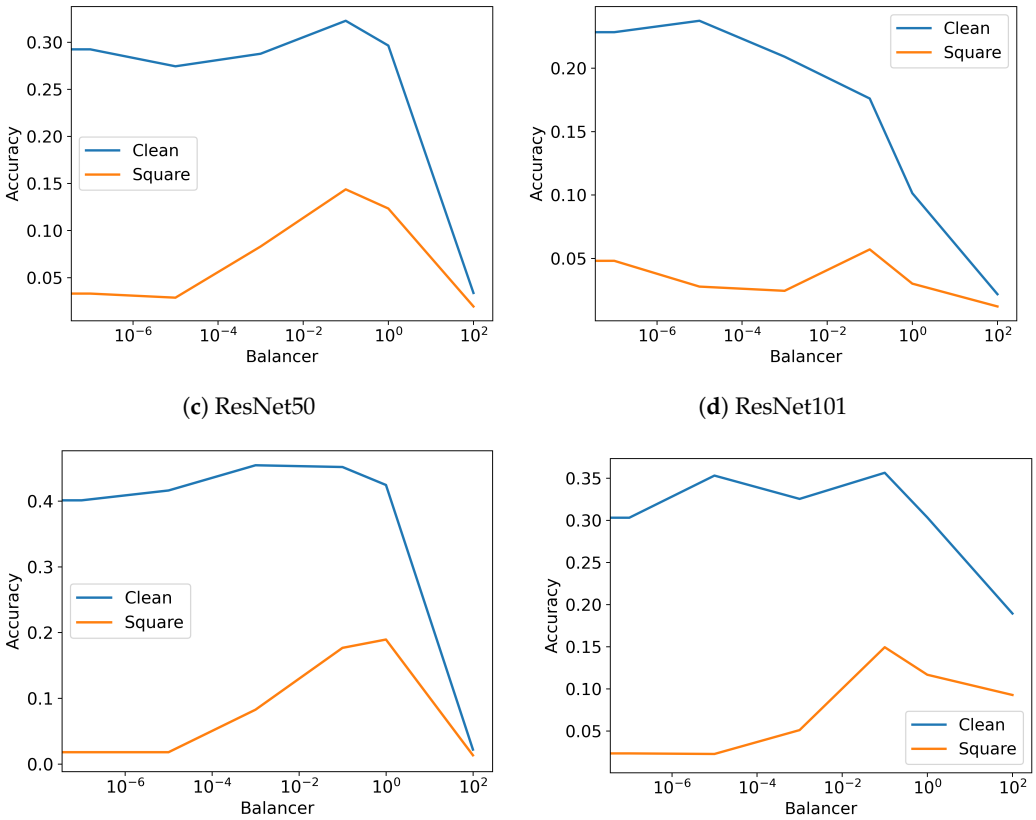


(**a**) Dense



(**b**) Shallow CNN

Figure 24. *Cont*.



(**c**) ResNet50 (**d**) ResNet101

(**e**) MobilenetV2 (**f**) InceptionV3

Figure 24. Accuracy of several types of networks on clean TinyImagenet and adversarial examples generated by black-box attacks when adding a dense layer with a D-ReLU function before the output layer and training them with augmented data samples generated from the EDM.

Table 9 presents a comparative analysis between the D-ReLU and TRADES methodologies, utilizing samples generated from the EDM approach while assessing the performance under a black-box attack across three distinct datasets: CIFAR10, CIFAR100, and TinyImagenet. In smaller network configurations such as those typified by the Dense and Shallow CNN architectures, the results observed under a black-box attack align closely with those obtained under white-box attacks, indicating consistent behavior across different types of adversarial attacks in these simpler network models. This consistency is crucial for validating the robustness of training methodologies against varied adversarial strategies.

Expanding the evaluation to deeper network architectures, particularly within the CIFAR10 and CIFAR100 datasets, D-ReLU demonstrates commendable competitiveness with TRADES. This indicates that D-ReLU can effectively leverage the complexities inherent in larger and deeper models to enhance robustness against black-box attacks, thereby suggesting its suitability in scenarios where maintaining integrity against external manipulations in data is critical.

Interestingly, in the TinyImagenet dataset, which typically requires handling of a more extensive and complex set of classes and image variations, D-ReLU not only competes well but also noticeably outperforms TRADES. This superior performance underscores D-ReLU's potential advantage in more challenging and diverse datasets where the depth and complexity of the network can be turned into a strategic asset to counter adversarial attacks more effectively.

Table 9. Accuracy for multiple types of networks under various robust training schemes with generated samples from the EDM, evaluated on both clean samples and adversarial examples generated by a black-box attack (i.e., square) on the CIFAR10, CIFAR100, and TinyImagenet datasets. Note that the accuracy metrics in bold are the highest in a specific model among the different training methods, and the numbers in parentheses are the ranks for training methods under an architecture.

Model	Training	CIFAR10		CIFAR100		TinyImagenet	
		Clean %	Square %	Clean %	Square %	Clean %	Square %
Dense	D-ReLU	52.6 (2)	**48.77** (1)	22.9 (2)	12.4 (2)	1.3 (2)	0.7 (2)
	TRADES	**62.47** (1)	47.23 (2)	**36.03** (1)	**23.6** (1)	**2.4** (1)	**0.93** (1)
Shallow CNN	D-ReLU	67.97 (2)	52.17 (2)	35.3 (2)	14.43 (2)	2.67 (2)	0.5 (2)
	TRADES	**74.3** (1)	**60.9** (1)	**44.23** (1)	**31.2** (1)	**7.33** (1)	**3.13** (1)
ResNet50	D-ReLU	79.1 (2)	64.93 (2)	53.83 (2)	33.03 (2)	**32.27** (1)	**14.37** (1)
	TRADES	**80.6** (1)	**67.9** (1)	**55.33** (1)	**40.07** (1)	7.33 (2)	4.7 (2)
ResNet101	D-ReLU	76.77 (2)	59.5 (2)	47.43 (2)	31.3 (2)	**17.6** (1)	**5.7** (1)
	TRADES	**77.97** (1)	**64.53** (1)	**52.6** (1)	**37.97** (1)	7.3 (2)	3.87 (2)
MobilenetV2	D-ReLU	**81.8** (1)	62.33 (2)	**56.57** (1)	31.27 (2)	**42.43** (1)	**18.93** (1)
	TRADES	79.33 (2)	**64.53** (1)	51.27 (2)	**37.97** (1)	18.13 (2)	9.9 (2)
InceptionV3	D-ReLU	87.4 (2)	74.73 (2)	**63.47** (1)	42.37 (2)	**35.63** (1)	**14.93** (1)
	TRADES	**87.73** (1)	**76.63** (1)	62.9 (2)	**48.8** (1)	12.2 (2)	6.63 (2)

8.2. Discussion

In the context of the CIFAR10 and CIFAR100 datasets, the integration of generated samples from the EDM approach appears to notably enhance the performance and robustness of both the D-ReLU and TRADES training methodologies. This improvement is primarily due to the diversification of data samples provided by EDM, which broadens the array of scenarios that the models encounter during training. Such enhanced variety promotes better generalization capabilities within machine learning models, equipping them to handle a wider range of inputs and reducing overfitting on the training data.

Furthermore, D-ReLU demonstrates a capacity to surpass TRADES in several state-of-the-art (SOTA) networks deployed on these datasets. This superior performance of D-ReLU suggests that its mechanisms might be more effectively aligned with the innate characteristics and challenges presented by the CIFAR10 and CIFAR100 datasets when combined with the enriched diversity of training instances generated through the EDM.

However, the scenario shifts quite dramatically when considering the TinyImagenet dataset. Both D-ReLU and TRADES exhibit significantly diminished performance compared to methodologies that do not employ EDM-generated samples. The core issue stems from the EDM's inability to produce new samples that accurately reflect the distribution inherent to the test dataset of TinyImagenet. The discrepancy between the training data augmented by the EDM and the actual data distribution encountered in testing hinders the model's ability to generalize effectively, resulting in poorer performance.

Despite these challenges with the TinyImagenet dataset, it is notable that D-ReLU still maintains a considerable performance edge over TRADES. This indicates that while the overall effectiveness of both methodologies is compromised by the limitations of the EDM in this context, D-ReLU's approach still manages to adapt more successfully than TRADES, leveraging its strengths to achieve better results even under less-than-ideal conditions.

Such findings underscore the importance of contextual suitability of data augmentation techniques like the EDM in training robust machine learning models. While the EDM proves advantageous in datasets like CIFAR10 and CIFAR100 by enhancing model generalization through diverse examples, its effectiveness is contingent upon the relevance and fidelity of the generated samples to the test environments. Tailoring the choice of augmentation strategies to the specific characteristics of the dataset is crucial in optimiz-

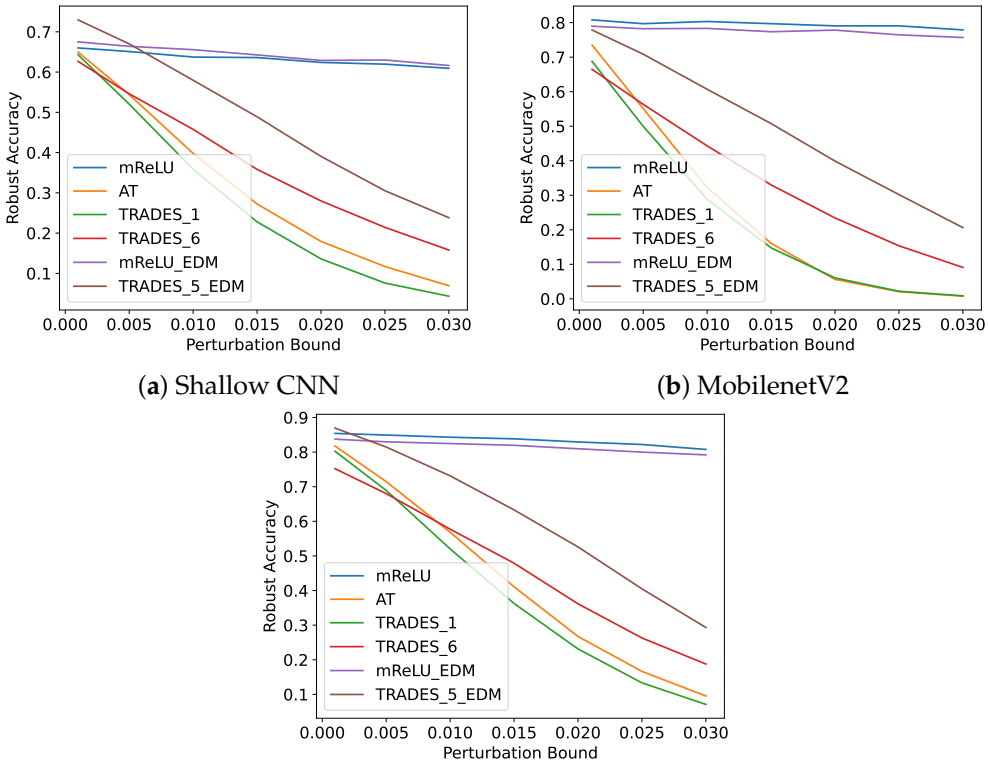
ing model performance and robustness. This nuanced approach to training can significantly influence the successful deployment of machine learning models across various real-world applications.

9. Perturbation-BoundGeneralization

This section demonstrates how D-ReLU and other baseline methods perform across various perturbation bounds. We choose APGE_CE as the adversarial attack in this experiment because it is the most widely used and one of the strongest attacks.

9.1. Experimental Results

Figure 25 presents the accuracy of various approaches, including the baselines and our proposed methods, on the CIFAR10 dataset under an APGD_CE attack with different levels of perturbation. For a small network like Shallow CNN, our approaches, D-ReLU and D-ReLU with the EDM outperform the other baselines under very small perturbations, except TRADES-5 with the EDM. However, as the perturbation level increases, D-ReLU and D-ReLU with the EDM consistently surpass all the baselines, demonstrating their superior robustness.

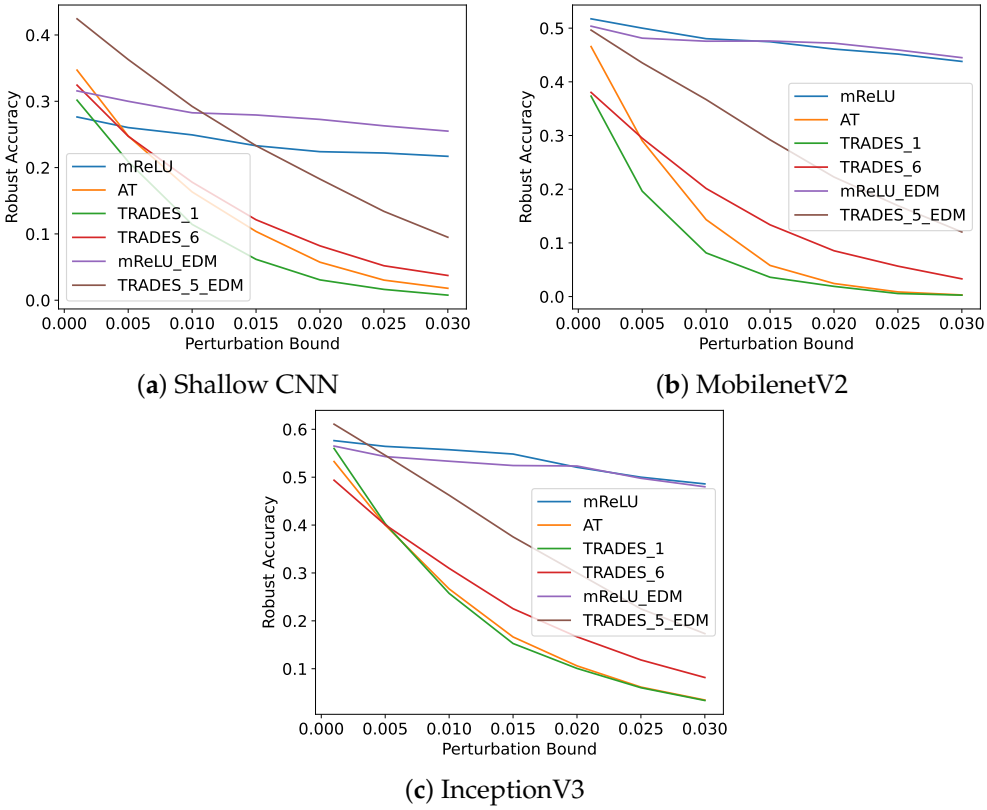


(**a**) Shallow CNN (**b**) MobilenetV2

(**c**) InceptionV3

Figure 25. Accuracy of several approaches on the CIFAR10 dataset under an APGD_CE attack with various perturbation bounds, where mReLU is D-ReLU.

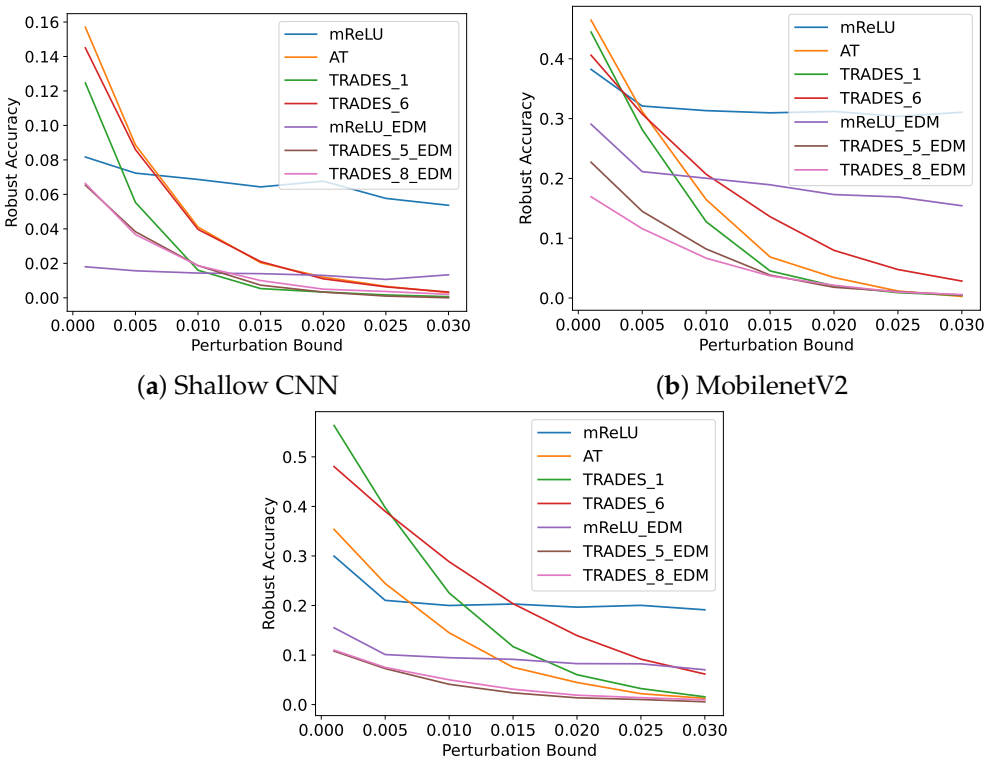
Figures 26 and 27 depict similar results for the CIFAR100 and TinyImagenet datasets, respectively. We observe a comparable trend to that of the CIFAR10 dataset, where D-ReLU and D-ReLU with the EDM exhibit enhanced performance over the baselines. Although our approaches show slightly diminished performance on larger datasets, they still generalize well across different perturbation bounds. This consistency across varying perturbation levels highlights our methods' robustness and adaptability.



(**a**) Shallow CNN (**b**) MobilenetV2

(**c**) InceptionV3

Figure 26. Accuracy of several approaches on the CIFAR100 dataset under an APGD_CE attack with various perturbation bounds, where mReLU is D-ReLU.



(**a**) Shallow CNN (**b**) MobilenetV2

(**c**) InceptionV3

Figure 27. Accuracy of several approaches on the TinyImagenet dataset under an APGD_CE attack with various perturbation bounds, where mReLU is D-ReLU.

9.2. Discussion

Our approaches, D-ReLU and D-ReLU with the EDM, demonstrate significant improvements in accuracy and robustness compared to baseline methods across different datasets and perturbation levels. These results indicate the potential of our techniques to enhance the reliability of machine learning models in adversarial settings, particularly in image classification tasks. Our methods maintain high accuracy under small perturbations and exhibit strong generalization capabilities as the perturbation bound increases, proving their effectiveness in real-world applications where robustness is critical.

10. Limitations

Despite the successful results of D-ReLU, this activation function may be more difficult than ReLU to harness because it has two hyperparameters. The first one is the balancer that was tuned in our experiments. Noticeably, the best balancer in the CIFAR10 dense network is different from the that in the CIFAR10 MobilenetV2 network. Therefore, it is tricky to find the best balancer. Moreover, the second hyperparameter is the initial max value of D-ReLU. We set it to 100 for the MNIST, CIFAR10, CIFAR100, and TinyImagenet datasets. It is clear that the results of our approach on the TinyImagenet are not very satisfying due to the large values before the D-ReLU layer, which cause several areas of zero gradients for training. Therefore, in large datasets, we may need to set it to a higher value. However, the results with an initial max value of 100 are satisfactory. It is noteworthy that if this value is ridiculously high, the training time will significantly increase because the optimizer takes much more time to reduce this max value.

11. Broader Impact

Our research is substantial, offering a transformative solution to the problem of adversarial vulnerability in machine learning systems by customizing activation functions within the model architecture. This enhancement in security was designed to be achieved without significantly affecting the model's performance on clean, non-adversarial samples. This is a critical advantage for machine learning practitioners who need to ensure that the pursuit of robustness does not come at the expense of efficiency and overall model accuracy.

The potential applications of this technology extend far beyond academic research; it has practical, real-world implications across various sectors utilizing artificial intelligence. Industries ranging from finance and healthcare to autonomous vehicle technology and cybersecurity can greatly benefit from the integration of our findings into their AI development cycles. By implementing our advanced techniques, these sectors can enhance the reliability and security of their systems against adversarial attacks, safeguarding sensitive data and critical operational functions.

Furthermore, our approach is expected to set a significant precedent for future research and development in adversarial robustness. By providing a versatile framework that can be adapted to diverse AI models and applications, our methodology promises to serve as a strong baseline for ongoing efforts in the mitigation of adversarial examples. Researchers and developers can leverage our proven strategies to explore further innovations in the field, potentially leading to even more sophisticated defenses against increasingly complex adversarial attacks.

Finally, the broader impacts of this research are multi-faceted, providing not only a practical method for enhancing the adversarial robustness of machine learning models but also contributing to the elevation of standards for the trustworthiness and security of AI systems in industrial applications. This work supports the important goal of advancing technology that is both powerful and resistant to evolving threats, thereby fostering a safer and more reliable digital future.

12. Conclusions and Future Works

We introduced the D-ReLU function to overcome the gradient vanishing issue observed with S-ReLU. We conducted various experiments demonstrating that D-ReLU

enhances adversarial robustness in larger datasets than MNIST. The results indicate that D-ReLU not only performed well but, in some instances, surpassed or matched the performance of TRADES under both white-box and black-box attack scenarios. Our statistical tests on the CIFAR10 dataset also show that D-ReLU significantly outperforms the other baselines.

Moreover, even when testing with augmented samples from the EDM, D-ReLU continued to show superior performance or remained competitive with TRADES. Notably, D-ReLU exhibited robust generalization across various perturbation bounds, a feature that TRADES struggled with. Integrating D-ReLU into a machine learning model offers a favorable balance between performance and robustness, making it a compelling option for enhancing model resilience against adversarial attacks.

In the future, we plan to design and implement a series of controlled experiments aimed at systematically evaluating how different initial maximum settings influence the performance and robustness of machine learning models, especially when applied to large-scale datasets. By manipulating this parameter, we aim to uncover deeper insights into how subtle changes can improve or impair a model's ability to withstand adversarial attacks, thereby refining the robustness of the activation function.

The anticipated outcome of these future investigations is a more nuanced understanding of the relationship between hyperparameters of the D-ReLU and the overall efficacy of the model. This will not only contribute to the academic literature but also provide practical guidelines that can be applied to enhance the security and reliability of machine learning systems in real-world applications. Through rigorous experimentation and analysis, we believe these efforts will pave the way for the development of more sophisticated, adaptive, and resilient machine learning architectures.

Author Contributions: Conceptualization, K.S. and P.R.; methodology, K.S. and P.R.; formal analysis, K.S. and P.R.; investigation, K.S. and K.S.; resources, K.S. and P.R.; data curation, K.S.; writing—original draft preparation, K.S.; writing—review and editing, K.S. and P.R.; visualization, K.S.; supervision, P.R.; project administration, P.R.; funding acquisition, P.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research was executed while P.R. and K.S. were funded by the National Science Foundation under grants NSF CISE—CNS Award 2136961 and 2210091.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Proof of Theorem 1

Proof. Suppose that we have a feedforward network. o_i^l denotes the output of neuron i in layer l, and w_{ij}^l is the parameter from neuron i in layer l to neuron j in layer $l+1$. Then, the output of neuron j in layer l with an activation function (denoted by $\mathrm{act}(\cdot)$) is

$$o_j^l = \mathrm{act}\left(\sum_i w_{ij}^{l-1} \cdot o_i^{l-1}\right).$$

When a previous layer has some perturbations (i.e., δ^{l-1}), the output is

$$\begin{aligned} o_j^{l^*} &= \mathrm{act}\left(\sum_i w_{ij}^{l-1} \cdot (o_i^{l-1} + \delta_i^{l-1})\right) \\ &= \mathrm{act}\left(\underbrace{\sum_i w_{ij}^{l-1} \cdot o_i^{l-1}}_{A} + \underbrace{\sum_i w_{ij}^{l-1} \cdot \delta_i^{l-1}}_{B}\right), \end{aligned}$$

where o^* means that the output has a perturbation induced by the previous layers.

Say that $A = \sum_i w_{ij}^{l-1} \cdot o_i^{l-1}$ and $B = \sum_i w_{ij}^{l-1} \cdot \delta_i^{l-1}$. Then, $o_j^l = \text{act}(A)$, and $o_j^{l^*} = \text{act}(A + B)$. Suppose that we would like to compare the differences between o_j^l and $o_j^{l^*}$ of ReLU and S-ReLU functions. Six cases can happen as follows:

- Case 1: $A \leq 0$ and $A + B > m \rightarrow |o_j^l - o_j^{l^*}| = |0 - (B - A)| = |A - B|$ for ReLU and $|o_j^l - o_j^{l^*}| = |0 - m| = |m|$ for S-ReLU. The perturbations in the output of S-ReLU are smaller than ReLU because $m < |A + B| < |A - B|$. The inequality is true, since A is negative and B is positive due to the conditions.
- Case 2: $0 < A \leq m$ and $A + B > m \rightarrow |o_j^l - o_j^{l^*}| = |A - (A + B)| = |B|$ for ReLU and $|o_j^l - o_j^{l^*}| = |A - m| = |A - m|$ for S-ReLU. The perturbations in the output of S-ReLU are smaller than those of ReLU because $B > m - A$ according to the conditions. Also, since both B and $m - A$ are positive due to the conditions, $|B| > |A - m|$.
- Case 3: $A > m$ and $A + B > m \rightarrow |o_j^l - o_j^{l^*}| = |A - (A + B)| = |B|$ for ReLU and $|o_j^l - o_j^{l^*}| = |m - m| = 0$ for S-ReLU. The perturbations in the output of S-ReLU are smaller than that of ReLU because $|B| > 0$.
- Case 4: $A > m$ and $0 < A + B \leq m \rightarrow |o_j^l - o_j^{l^*}| = |A - (A + B)| = |B|$ for ReLU and $|o_j^l - o_j^{l^*}| = |m - (A + B)| = |B + A - m|$ for S-ReLU. The perturbations in the output of S-ReLU are smaller than ReLU because $A - m$ is positive due to the conditions, and B is negative. Then, $B + A - m$ is greater than B. Thus, $|B + A - m|$ is less than $|B|$.
- Case 5: $A > m$ and $A + B \leq 0 \rightarrow |o_j^l - o_j^{l^*}| = |A - 0| = |A|$ for ReLU and $|o_j^l - o_j^{l^*}| = |m - 0| = |m|$ for S-ReLU. The perturbations in the output of S-ReLU are smaller than that of ReLU because one of the conditions is $A > m$. Then, $|m| < |A|$.
- Case 6: $A \leq m$ and $A + B \leq m \rightarrow |o_j^l - o_j^{l^*}|$ for both ReLU and S-ReLU because S-ReLU behaves the same as ReLU.

These results are summarized in Table A1 and show that the output of S-ReLU never exceeds that of ReLU. Therefore, the theorem is valid. □

Table A1. The difference between the outputs of a layer in a model on a clean sample and a sample injected by small perturbations under possible conditions.

Conditions	Output Difference	
	ReLU	S-ReLU
$A \leq 0$ and $A + B > m$	$\|A - B\|$	$\|m\|$
$0 < A \leq m$ and $A + B > m$	$\|B\|$	$\|A - m\|$
$A > m$ and $A + B > m$	$\|B\|$	0
$A > m$ and $0 < A + B \leq m$	$\|B\|$	$\|B + A - m\|$
$A > m$ and $A + B \leq 0$	$\|A\|$	$\|m\|$
$A \leq m$ and $A + B \leq m$	Same	

Appendix B. Proof of Corollary 1

Proof. This corollary can be easily proven by the information in Table A1 summarized from the proof of Theorem 1. When m decreases, S-ReLU's $|o_j^l - o_j^{l^*}|$ also decreases or remains the same. For example, in case 3, suppose that $m' < m$. Therefore, $|B + A - m'| < |B + A - m|$ because $B + A \leq m$ and $B + A \leq m'$ according to the condition. □

References

1. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing properties of neural networks. *arXiv* **2013**, arXiv:1312.6199.
2. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. *arXiv* **2014**, arXiv:1412.6572.
3. Sooksatra, K.; Hamerly, G.; Rivas, P. Is ReLU Adversarially Robust? In Proceedings of the LatinX in AI Workshop at ICML 2023, Honolulu, HI, USA, 23–29 July 2023. [CrossRef]
4. Kurakin, A.; Goodfellow, I.; Bengio, S. Adversarial examples in the physical world. *arXiv* **2016**, arXiv:1607.02533.

5. Papernot, N.; McDaniel, P.; Jha, S.; Fredrikson, M.; Celik, Z.B.; Swami, A. The limitations of deep learning in adversarial settings. In Proceedings of the 2016 IEEE European Symposium on Security and Privacy (EuroS&P), Saarbruecken, Germany, 21–24 March 2016; pp. 372–387.
6. Carlini, N.; Wagner, D. Towards evaluating the robustness of neural networks. In Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2017; pp. 39–57.
7. Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; Vladu, A. Towards deep learning models resistant to adversarial attacks. *arXiv* **2017**, arXiv:1706.06083.
8. Ilyas, A.; Engstrom, L.; Athalye, A.; Lin, J. Black-box adversarial attacks with limited queries and information. In Proceedings of the International Conference on Machine Learning, PMLR, Stockholm, Sweden, 10–15 July 2018; pp. 2137–2146.
9. Sooksatra, K.; Rivas, P. Enhancing Adversarial Examples on Deep Q Networks with Previous Information. In Proceedings of the 2021 IEEE Symposium Series on Computational Intelligence (SSCI), Orlando, FL, USA, 5–7 December 2021; pp. 1–7.
10. Meng, D.; Chen, H. Magnet: A two-pronged defense against adversarial examples. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, Dallas, TX, USA, 30 October–3 November 2017; pp. 135–147.
11. Samangouei, P.; Kabkab, M.; Chellappa, R. Defense-GAN: Protecting Classifiers Against Adversarial Attacks Using Generative Models. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
12. Cohen, J.; Rosenfeld, E.; Kolter, Z. Certified adversarial robustness via randomized smoothing. In Proceedings of the International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 1310–1320.
13. Papernot, N.; McDaniel, P.; Wu, X.; Jha, S.; Swami, A. Distillation as a defense to adversarial perturbations against deep neural networks. In Proceedings of the 2016 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2016; pp. 582–597.
14. Carlini, N.; Wagner, D. Defensive distillation is not robust to adversarial examples. *arXiv* **2016**, arXiv:1607.04311.
15. Wong, E.; Kolter, Z. Provable defenses against adversarial examples via the convex outer adversarial polytope. In Proceedings of the International Conference on Machine Learning, PMLR, Stockholm, Sweden, 10–15 July 2018; pp. 5286–5295.
16. Pang, T.; Du, C.; Dong, Y.; Zhu, J. Towards robust detection of adversarial examples. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 4579–4589.
17. Liu, J.; Zhang, W.; Zhang, Y.; Hou, D.; Liu, Y.; Zha, H.; Yu, N. Detection based defense against adversarial examples from the steganalysis point of view. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4825–4834.
18. Zhao, Z.; Chen, G.; Wang, J.; Yang, Y.; Song, F.; Sun, J. Attack as defense: Characterizing adversarial examples using robustness. In Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis, Virtual Event, 11–17 July 2021; pp. 42–55.
19. Nesti, F.; Biondi, A.; Buttazzo, G. Detecting adversarial examples by input transformations, defense perturbations, and voting. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *34*, 1329–1341. [CrossRef] [PubMed]
20. Li, Q.; Chen, J.; He, K.; Zhang, Z.; Du, R.; She, J.; Wang, X. Model-agnostic Adversarial Example Detection via High-Frequency Amplification. In *Computers & Security*; Elsevier: Amsterdam, The Netherlands, 2024; p. 103791.
21. Rakin, A.S.; Yi, J.; Gong, B.; Fan, D. Defend deep neural networks against adversarial examples via fixed and dynamic quantized activation functions. *arXiv* **2018**, arXiv:1807.06714.
22. Qian, H.; Wegman, M.N. L2-nonexpansive neural networks. *arXiv* **2018**, arXiv:1802.07896.
23. Xie, C.; Wu, Y.; van der Maaten, L.; Yuille, A.L.; He, K. Feature denoising for improving adversarial robustness. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 501–509.
24. Sehwag, V.; Wang, S.; Mittal, P.; Jana, S. Hydra: Pruning adversarially robust neural networks. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 19655–19666.
25. Wu, B.; Chen, J.; Cai, D.; He, X.; Gu, Q. Do wider neural networks really help adversarial robustness? *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 7054–7067.
26. Chen, X.; Li, X.; Zhou, Y.; Yang, T. DDDM: A Brain-Inspired Framework for Robust Classification. In Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI-22), Vienna, Austria, 23–29 July 2022.
27. Bubeck, S.; Li, Y.; Nagaraj, D.M. A law of robustness for two-layers neural networks. In Proceedings of the Conference on Learning Theory, PMLR, Boulder, CO, USA, 15–19 August 2021; pp. 804–820.
28. Bubeck, S.; Sellke, M. A universal law of robustness via isoperimetry. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 28811–28822. [CrossRef]
29. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4510–4520.
30. Deng, L. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Process. Mag.* **2012**, *29*, 141–142. [CrossRef]
31. Krizhevsky, A.; Hinton, G. *Learning Multiple Layers of Features from Tiny Images*; University of Toronto: Toronto, ON, Canada, 2009.
32. Le, Y.; Yang, X.S. *Tiny ImageNet Visual Recognition Challenge*; 2015.
33. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.

34. Croce, F.; Hein, M. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In Proceedings of the International Conference on MACHINE learning, PMLR, Virtual Event, 13–18 July 2020; pp. 2206–2216.
35. Andriushchenko, M.; Croce, F.; Flammarion, N.; Hein, M. Square attack: A query-efficient black-box adversarial attack via random search. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2020; pp. 484–501.
36. Zhang, H.; Yu, Y.; Jiao, J.; Xing, E.; El Ghaoui, L.; Jordan, M. Theoretically principled trade-off between robustness and accuracy. In Proceedings of the International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 7472–7482.
37. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
38. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
39. Demšar, J. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* **2006**, *7*, 1–30.
40. Wang, Z.; Pang, T.; Du, C.; Lin, M.; Liu, W.; Yan, S. Better diffusion models further improve adversarial training. In Proceedings of the International Conference on Machine Learning, PMLR, Honolulu, HI, USA, 23–29 July 2023; pp. 36246–36263.
41. Karras, T.; Aittala, M.; Aila, T.; Laine, S. Elucidating the design space of diffusion-based generative models. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 26565–26577.