ABSTRACT

Safe and Robust Neural Architectures via Limiting the Activation Potential of Neurons with Rectified Linear Units

Korn Sooksatra, Ph.D.

Mentor: Pablo Rivas, Ph.D.

The widespread use of deep learning models in critical fields such as healthcare and autonomous transportation underlines the necessity for robust security against adversarial examples — inputs deliberately modified to mislead models. Despite superior performance in many tasks, these models are vulnerable to attacks that compromise safety. Previous defense strategies, including adversarial training and gradient masking, have proven either computationally intense or partially effective. This research targets the susceptibility inherent in the ReLU activation functions, proposing custom modifications intended to bolster model defense without affecting performance. Our evaluations across various datasets indicate improved robustness, showcasing the efficacy of these architectural enhancements in mitigating adversarial vulnerabilities.

Hold for signature page

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my advisor, Dr. Pablo Rivas, for his invaluable guidance, unwavering support, and mentorship throughout the duration of this dissertation. His insights and advice have been instrumental in shaping both the direction and success of this research. His dedication to excellence and scholarly rigor have profoundly inspired and motivated me. I am also incredibly thankful to Dr. Greg Hamerly for his advisory role at the initial stage of this dissertation. His early contributions to the conceptual framework of this research were crucial and have left a lasting impact on the final outcome. My heartfelt thanks extend to all the members of my dissertation committee. In particular, I want to recognize Dr. Pablo Rivas, Dr. Greg Hamerly, Dr. Mary Lauren Benton, and Dr. Liang Dong for their valuable feedback, responsive engagement, and flexibility throughout the research process. Their collective expertise and thoughtful criticism have been fundamental to my developmental journey as a researcher. Additionally, I am indebted to my labmates, including Bikram Khanal, Ernesto Quevedo Caballero, Maisha Binte Rashid, Jorge Yero Salazar, Alejandro Rodriguez Perez, and many others. Sharing a lab space with such talented individuals made it possible for me to conduct multiple important experiments and provided a stimulating and enriching environment that was crucial for my work. The collaboration, camaraderie, and shared curiosity amongst us have made my research experience not only productive but also extraordinarily enjoyable. I would also like to acknowledge the exceptional support from Sharon Humphrey, Candace Ditsch, and Dr. Eunjee Song in the administrative and procedural aspects related to graduation. Their guidance through the complexities of the graduation process was indispensable, ensuring a smooth culmination to my doctoral studies. Each of you has contributed to this journey in unique and significant ways, and for

that, I am eternally grateful. Thank You for being a part of this pivotal chapter of my academic life.

This work was partially supported by the National Science Foundation under Grant 2210091. The views expressed herein are solely those of the author and do not necessarily reflect those of the National Science Foundation.

This dissertation is dedicated to my beloved wife, Pilaiporn Phetcherdchin, whose unwavering support and encouragement have kept me motivated throughout this journey, and to my wonderful parents, Somboon Sooksatra and Vannee Sooksatra, and my loving grandmother, Uraiporn Pongwarin, whose steadfast support and belief in my abilities have been the foundation of my educational success.

ATTRIBUTIONS

Chapter Three shows the limitations of ReLU activation functions, a problem first identified by Dr. Greg Hamerly, who recognized their potential role in the emergence of adversarial examples. Dr. Pablo Rivas provided a thorough review and revision to prepare this chapter for publication.

Chapter Four presents the design and implementation of the S-ReLU activation function, an enhancement of the ReLU function to improve adversarial robustness. The zero gradient experiment, proposed by Dr. Greg Hamerly, was instrumental in validating the effectiveness of S-ReLU. Dr. Pablo Rivas recommended evaluating S-ReLU using MNIST-like datasets, as it is specifically suited for small-scale datasets. Additionally, Dr. Rivas provided a comprehensive review and revision to prepare the chapter for publication.

CHAPTER ONE

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, some of which 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—subtle and deliberately engineered modifications to input data crafted to mislead the model into making erroneous decisions. Figure 1.1 illustrates an adversarial example that can fool an image classifier from predicting the image as a dog to predicting the image as a cat. This susceptibility was first identified and discussed in seminal papers (Szegedy et al., 2013; Goodfellow et al., 2014), highlighting

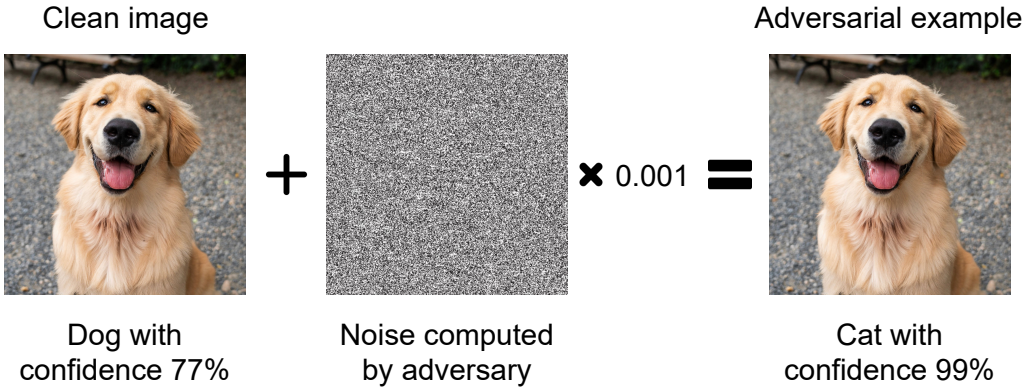


Figure 1.1. Adversarial example that misleads an image classifier to predict this image as a cat.

significant challenges in deploying these models in environments demanding high security and reliability.

Addressing the vulnerabilities posed by adversarial examples has led to a plethora of research endeavors (Goodfellow et al., 2014; Kurakin et al., 2016; Papernot et al., 2016; Carlini and Wagner, 2017; Madry et al., 2017; Ilyas et al., 2018; Sooksatra and Rivas, 2021). Among the proposed solutions, adversarial training has emerged as a foremost strategy due to its relatively straightforward implementation and proven effectiveness (Madry et al., 2017). 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 auto-encoders and generative adversarial networks (GANs) has been explored to preprocess and potentially cleanse adversarial perturbations from inputs (Meng and Chen, 2017; Samangouei et al., 2018). These methods aim to improve the robustness of machine learning models by denoising or altering the input data before the model processes it. Figure 1.2 demonstrates the autoencoder method, which preprocesses an input to ensure that the classifier or target model receives a clean input. This autoencoder has been specifically trained to denoise adversarial examples, effectively reducing the impact of adversarial perturbations.

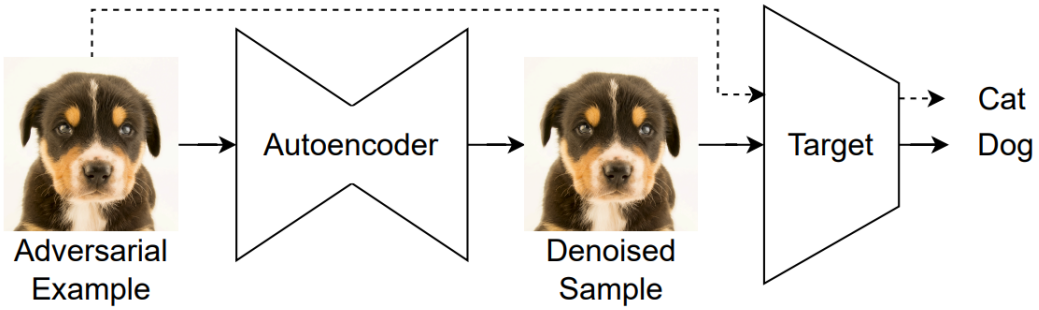


Figure 1.2. Denoised autoencoder for preprocessing an adversarial example to create a clean/denoised sample. The solid line is the process with the autoencoder, and the dashed line is the process without the autoencoder.

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.

In addition, some strategies have focused on leveraging the outputs from machine learning models to promote robustness. One notable approach is randomized smoothing (Cohen et al., 2019), 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 1.3 illustrates the randomized smoothing method. In this example, the system generates four predictions: three predict the inputs as a dog, and one predicts it as a cat. Based on the majority vote, the final output of the system is a dog.

Another method, defensive distillation, aims to reduce a model's sensitivity to input variations by training the model to output softened probabilities rather than

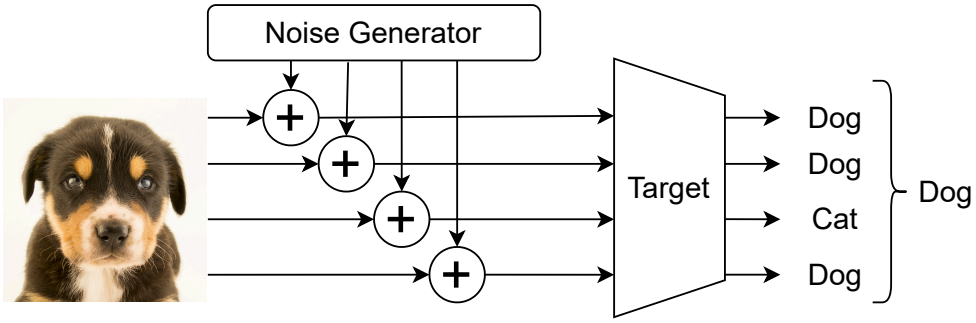


Figure 1.3. Randomized smoothing method where most predictions are picked as the output. In this example, four noises are generated from the noise generator.

hard classifications (Papernot et al., 2016). Despite its initial promise, defensive distillation is vulnerable to more sophisticated adversarial attacks, as demonstrated by (Carlini and Wagner, 2016). 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 (Wong and Kolter, 2018; Pang et al., 2018; Liu et al., 2019; Zhao et al., 2021; Nesti et al., 2021; Li et al., 2024). 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 1.4 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.

All the techniques above focus on preprocessing, post-processing, or augmenting the inputs and outputs rather than directly modifying the models' architectures.

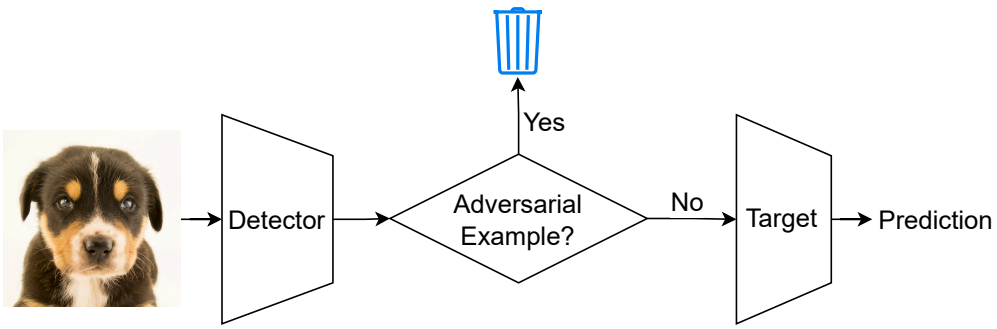


Figure 1.4. Adversarial example detection technique where the detected samples are thrown away.

However, the architecture of the models, mainly the activation functions, significantly contributes to their vulnerability to adversarial examples. As illustrated in Figure 1.1, 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 (Rakin et al., 2018; Qian and Wegman, 2018; Xie et al., 2019; Sehwag et al., 2020; Wu et al., 2021; Chen et al., 2022; Bubeck et al., 2021; Bubeck and Sellke, 2021) has aimed to customize the models' architectures, only a few works have specifically targeted the customization of activation functions. One such effort by (Rakin et al., 2018) 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 (Sandler et al., 2018), which caps the activation values at 6. Although ReLU6 was introduced to improve robustness, more exploration of its potential must be explored to enhance robustness against adversarial attacks.

Amidst these challenges, we observe a potential opportunity in the conventional solutions—the activation functions within deep learning architectures, precisely the ReLU function, which is known to contribute to vulnerabilities against adversarial examples Goodfellow et al. (2014). 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.

1.1 Objectives

This dissertation aims to address the vulnerability of deep learning models to adversarial examples by enhancing the robustness of activation functions. Specifically,

we focus on refining the ReLU function to mitigate the effects of adversarial perturbations. The objectives of this research are as follows:

1.1.1 Enhance the Robustness of Machine Learning Models Without Significant Detriment to Accuracy

We propose customizing the ReLU activation function to minimize perturbation effects and diminish gradients, thereby promoting a more secure model operation. This customized function, the static-max-value ReLU (S-ReLU), is designed to resist adversarial perturbations more effectively than the traditional ReLU. We will:

- Develop and formalize the S-ReLU function.
- Conduct a theoretical analysis comparing the extent of perturbation that can pass through general ReLU and S-ReLU functions.
- Perform experiments on variants of the MNIST dataset (Deng, 2012) to validate the practical effectiveness of S-ReLU in enhancing model robustness.

1.1.2 Augment the Robustness of State-of-the-Art Pre-Trained Deep Learning Models for Safer Public Deployment

Building on the theoretical and empirical findings from the first objective, we design a new customized ReLU function called the dynamic-max-value ReLU (D-ReLU). This function is tailored to improve the robustness of models on larger datasets. We will:

- Design the D-ReLU function to adapt dynamically to input data.
- Integrate D-ReLU into popular deep learning models, incorporating custom ReLU modifications and additional dense layers where necessary.
- Conduct comprehensive experiments on larger datasets such as CIFAR-10, CIFAR-100, and TinyImagenet to evaluate the practical capabilities and robustness improvements D-ReLU offers.

These objectives aim to develop and validate new activation functions that enhance the security and reliability of deep learning models against adversarial attacks, thereby contributing to safer deployment in critical applications.

1.2 Contributions

This dissertation makes several significant contributions to deep learning, particularly in enhancing the robustness of machine learning models against adversarial examples through refining activation functions. The key contributions are as follows:

1.2.1 Design and Implementation of Static-Max-Value ReLU (S-ReLU) Function

- Innovation in Activation Functions: We introduce the S-ReLU function, a novel activation function designed to reduce the impact of adversarial perturbations by customizing the ReLU function to minimize perturbation effects and diminish gradients.
- Theoretical Analysis: We provide a comprehensive theoretical analysis comparing the robustness of the traditional ReLU function with the S-ReLU function, demonstrating how S-ReLU can more effectively limit the passage of perturbations.
- Experimental Validation: We validate the effectiveness of S-ReLU through extensive experiments on variants of the MNIST dataset, showcasing its ability to enhance model robustness without significantly compromising accuracy.

1.2.2 Design and Implementation of Dynamic-Max-Value ReLU (D-ReLU) Function

- Advanced Activation Function Design: We design the D-ReLU function, an advanced activation function that dynamically adjusts to input data, further enhancing model robustness on larger and more complex datasets.
- Integration with Pre-Trained Models: We integrate D-ReLU into state-of-the-art pre-trained deep learning models, demonstrating how custom ReLU

modifications and additional dense layers can improve model security and reliability.

- Comprehensive Experiments: We conduct a series of experiments on larger datasets, including CIFAR-10, CIFAR-100, and TinyImageNet, to evaluate the practical performance and robustness improvements provided by D-ReLU. These experiments highlight D-ReLU's potential for safer public deployment of deep learning models.

1.2.3 Framework for Evaluating and Comparing Traditional and Modified Activation Functions

- Evaluation Metrics: We establish a robust framework for evaluating the effectiveness of different activation functions in mitigating adversarial attacks, including theoretical and empirical metrics.
- Benchmarking Results: Our experimental results on various datasets and machine learning models' architectures serve as benchmarks for future research in developing robust activation functions, clearly comparing the performance and resilience of traditional and customized ReLU functions.

1.2.4 Practical Implications for Safe Deployment

- Guidelines for Model Deployment: Based on our findings, we propose practical guidelines for deploying deep learning models in security-critical applications, emphasizing the importance of robust activation functions in enhancing model safety.
- Contributions to Model Robustness: Our research improves the robustness and reliability of deep learning models, facilitating their adoption in fields where security and accuracy are paramount.

These contributions advance the understanding and application of activation functions in deep learning, offering new avenues for research and practical solutions for enhancing model robustness against adversarial threats.

1.3 Overview of Remaining Chapters

Chapter Two provides an in-depth review of existing literature on adversarial attacks and their corresponding defense mechanisms. It begins by outlining and examining previous studies and reviews of this emerging field. The chapter then elaborates on the taxonomy of adversarial defense techniques, cataloging them accordingly to help clarify the landscape of existing methodologies. Furthermore, it explores the evolutionary trends of these techniques and the datasets that have been predominantly utilized over the years. It provides a comprehensive understanding of the historical and current state of research in adversarial machine learning.

Chapter Three discusses the vulnerabilities associated with ReLU functions when facing adversarial examples. Through detailed experimentation and analysis, this chapter illustrates the susceptibility of ReLU functions to such attacks, reinforcing the arguments with empirical results. The experimental results are meticulously presented to corroborate our hypothesis about the weakness of ReLU functions and provide a solid foundation for exploring alternative solutions.

Chapter Four introduces a novel activation function named static-max-value ReLU (S-ReLU) designed to mitigate some vulnerabilities identified in traditional ReLU functions. This chapter begins with a theoretical analysis comparing the levels of adversarial perturbations that networks can sustain using traditional ReLU functions versus those employing S-ReLU. Following the theoretical analysis, a series of experiments are conducted to evaluate the performance of S-ReLU. The results of these experiments are discussed in detail, demonstrating improved adversarial robustness and overall performance enhancements in neural network models utilizing S-ReLU.

Chapter Five explores another innovative activation function termed dynamic-max-value ReLU (D-ReLU). This chapter introduces a novel loss function explicitly designed for training models with D-ReLU and details the procedural methodology for implementing such training. In addition, it presents a comparative analysis through several experimental setups to benchmark the new D-ReLU against other baseline approaches. The findings are extensively reviewed to underscore the advantages and potential limitations of using D-ReLU in machine learning models.

Chapter Six serves as the concluding segment of this dissertation, summarizing the essential findings and contributions of the research. It critically reflects on the work undertaken and proposes potential avenues for future research to refine the D-ReLU mechanism. The discussion focuses on enhancing the trainability and effectiveness of D-ReLU further to improve machine learning models' performance and adversarial robustness. This chapter seeks to inspire ongoing and future research in secure and robust artificial intelligence.

CHAPTER TWO

Literature Reviews

This section examines the comprehensive survey and focuses on the multitude of techniques utilized to enhance adversarial robustness in machine learning models. Throughout our discussion, we emphasize the methodologies employed, dissecting various strategies developed over the years to fend off or mitigate the effects of adversarial attacks on systems.

We will analyze commonly used datasets and related techniques in this field. Our review will trace the evolution of these datasets and defensive techniques over the years. This historical perspective will highlight emerging trends, how approaches have adapted to new adversarial tactics, and suggest future directions for adversarial robustness research.

We aim to highlight key milestones in developing these techniques and assess the effectiveness of various strategies in different contexts. By examining the progression of approaches, we can identify pivotal advancements, evaluate the current state of adversarial robustness techniques, and speculate on future research directions and potential breakthroughs.

This exploration enhances our understanding of adversarial robustness and serves as a guide for researchers and practitioners in selecting and implementing appropriate defensive mechanisms for specific challenges in cybersecurity and machine learning.

2.1 Existing Surveys

Chakraborty et al. (2018) laid the groundwork with a comprehensive survey covering various types of attacks and defense mechanisms in adversarial learning. They aimed to summarize recent findings and methodologies, providing practical

examples and insights into defense strategies. Building upon this foundation, Ren et al. (2020) expanded on categorizing attacks based on adversary knowledge and discussed defensive techniques such as adversarial training. Concurrently, Ren et al. (2021) went deeper into the exploration of adversarial examples, both in digital and physical realms, while Bai et al. (2021) reviewed past adversarial training techniques.

Wang et al. (2022) addressed the issue of adversarial samples deceiving DNN models, categorizing attacks and defenses. Similarly, Li et al. (2022) reviewed adversarial attack and defense techniques with practical demonstrations. Focusing on robust adversarial training, Qian et al. (2022) offered a systematic overview of methodologies. Complementing these, Li et al. (2023) examined certifiably robust defenses against evasion attacks, benchmarking existing methods. Vorobeychik (2023) provided a broad overview of adversarial machine learning, highlighting its evolution.

Additionally, Li and Li (2024) discussed the balance between accuracy, robustness, and fairness in machine learning models, exploring trade-offs. These surveys collectively offer a comprehensive understanding of adversarial learning challenges and advancements, guiding future research.

Building on this foundation, our review focuses specifically on defenses against adversarial examples in convolutional neural networks (CNNs) for image classification. Unlike broader surveys, we analyze the efficacy and limitations of defense mechanisms tailored for CNNs, given their widespread use in image applications. By presenting a taxonomy of defense strategies, we aim to provide actionable insights for researchers and practitioners in computer vision and image processing. This analysis contributes to enhancing the robustness and reliability of CNNs in real-world image classification tasks.

2.2 Taxonomy of Adversarial Defenses

Numerous studies have focused on enhancing the robustness of machine learning models. Our investigation revealed a variety of techniques aimed at achieving

adversarial robustness. These techniques can be classified into five distinct categories, each with its own unique approach: detection-based methods, training-based strategies, architecture-based approaches, preprocessing-based techniques, postprocessing-based solutions, and combination methods that leverage a mix of these approaches. By breaking down these techniques into categories, we can better understand the diverse range of options available for improving the resistance of machine learning models against adversarial attacks.

2.2.1 Detection-Based Techinque

This technique detects adversarial examples before they are fed into a machine learning model, whether with a machine-learning or non-machine-learning approach. Wong and Kolter (2018) introduced a provably robust training approach, ensuring classifier robustness through the construction of convex outer bound and efficient optimization. This convex outer bound means that they transform a ReLU function into a linear function by considering the lower and upper bounds because the ReLU function is difficult to compute. This method guarantees robustness against norm-bounded perturbations and provides insights into model performance and the ability to detect adversarial examples at test time.

In contrast, Pang et al. (2018) proposed a non-maximal entropy metric combined with a reverse classifier strategy to enhance robustness. The non-maximum entropy can be computed by

$$-\Sigma_{i\neq\hat{y}}\hat{F}(x)_i \log(\hat{F}(x)_i),$$

where x is an input, $F(x)$ is a classifier, $F(x)_i$ is the logit of class i, $\hat{y}$ is the predicted label, and $\hat{F}(x)_i = \frac{F(x)_i}{\sum_{j\neq\hat{y}} F(x)_j}$. By quantifying entropy in classifier outputs, they effectively differentiate normal inputs from adversarial ones, concealing normal examples in low-dimensional manifolds to thwart attacks. Similarly, Liu et al. (2019) addressed adversarial attacks by leveraging steganalysis to identify deviations caused

by perturbations, constructing feature-based detectors to discern adversarial examples. These methods focus on leveraging domain-specific characteristics and detection strategies to enhance robustness.

Furthermore, Zhao et al. (2021) designed attack cost-based detection approaches, considering the cost of attacks to differentiate between benign and adversarial examples. Their K-NN and Z-Score based methods demonstrate effectiveness and robustness in classification. Similarly, Nesti et al. (2021) proposed architectures tailored to detect various types of adversarial examples. First, they employed the baseline model to differentiate between a clean and an adversarial examples by altering an image with a number of transformations and feeding them to a target. At the end, if the KL divergence of the outputs of the target based on those images exceeded the specified threshold, then the image is an adversarial example.

Lastly, Li et al. (2024) introduced a frequency domain analysis approach inspired by adversarial perturbations, leveraging high-frequency signals and edge-based feature extraction for effective detection. Collectively, these approaches contribute to ongoing efforts in enhancing the robustness of machine learning models against adversarial attacks by offering diverse detection strategies and insights into model vulnerabilities.

2.2.2 Training-Based Techinque

This particular technique involves adjusting the training process in order to bolster the adversarial robustness of machine learning models. This modification can be further subdivided into three distinct categories, each presenting its own set of approaches and methodologies for improving the model's resistance to adversarial attacks. By going into these categories, we can gain a deeper understanding of how subtle adjustments during the training phase can significantly enhance the robustness of machine learning models in the face of potential adversarial threats.

2.2.2.1 Retraining. The literature on retraining for adversarial robustness encompasses various approaches, each contributing to the understanding and enhancement of classifier resilience against adversarial attacks.

Goodfellow et al. (2014) pioneered adversarial training with the Fast Gradient Sign Method (FGSM), demonstrating its efficacy in evaluating robustness by retraining classifiers with adversarial examples generated using FGSM. FGSM was a one-step method that could be computed by

$$x + \epsilon \cdot \text{sign}(\nabla_x l(F(x, \theta), y)),$$

where x is an input, y is its ground truth label, $F(x, \theta)$ is a classifier, θ is the classfier's parameters, ϵ is the perturbation bound or a budget of an adversary, $l(a, b)$ is a loss function of predicted a with respect to the label b, and $\text{sign}(\cdot)$ is a function that outputs the signs of its input. Building upon this, Madry et al. (2017) introduced Projected Gradient Descent (PGD) as a stronger attack. PGD was a multi-step method that was similar with FGSM and could be computed by

$$x + \eta \cdot \text{sign}(\nabla_x l(F(x, \theta), y)),$$

where η is a small step so that the attack could refine the adversarial slowly. Also, this equation would be computed multiple times until it found an adversarial example or exceeded its max iteration. The authors utilized this attack in their adversarial training. The training's goal was to solve this problem:

$$\min_\theta \max_{\delta \in \Gamma} l(F(x + \delta, \theta), y),$$

where x is an input, y is the label, θ is the parameters of the target, $F(x, \theta)$ is the target, δ is the adversarial perturbations, Γ is the perturbation bound, and $l(F(x, \theta), y)$ is a loss function. In general speaking, this optimization problem tried to minimize the worst possible case that could happen with the given perturbation bound Γ. The inner maximization was substituted with an adversarial attack like FGSM or PGD. Their work showcased the effectiveness of retraining classifiers with PGD-generated

adversarial examples, enhancing robustness against not only L_∞ attacks like FGSM and PGD but also against L_2 attacks.

Recognizing the limitations of adversarial training with examples from a single classifier, Tramèr et al. (2018) proposed Ensemble Adversarial Training, advocating for the use of adversarial examples from multiple classifiers to bolster robustness, particularly against black-box attacks. Shafahi et al. (2019) introduced Natural Gradient Adversarial Training, leveraging gradients from natural training to compute adversarial perturbations, albeit requiring extensive training epochs to achieve robustness comparable to PGD.

Tramer and Boneh (2019) delved into theoretical limits of adversarial robustness within a natural statistical model, exploring trade-offs between robustness and accuracy across various perturbation types. They proposed new adversarial training schemes tailored to multi-perturbation risks. In contrast, Wong et al. (2020) demonstrated the efficiency of FGSM alone in achieving robustness, suggesting potential optimizations in adversarial training procedures.

Through categorization, it becomes evident how different approaches to retraining for adversarial robustness have evolved, from specific attack methods like FGSM and PGD to ensemble methods and considerations of theoretical limits. Efficiency considerations, such as those presented by Wong et al. (2020), further contribute to the optimization of adversarial training procedures.

2.2.2.2 Regularization. Recent advancements in regularization-based adversarial robustness have explored various strategies to mitigate the accuracy-robustness tradeoff in deep learning models.

Zhang et al. (2019) introduced TRADES, which splits the objective function into two terms: one for accuracy and the other for robustness. The new objective function is shown here:

$$d(F(x), y) + \beta d(F(x), F(x^{'})),$$

where $d(F(x), y)$ is a distance function, which is generally the KL divergence, between $F(x)$ and y, $x^{'}$ is an adversarial example that can be found with PGD, and β is a balancer between the performance and robustness. The goal is to minimize these distances, and this approach enables finding a balance between accuracy (the first part) and robustness (the second part) during training.

Similarly, Wang et al. (2019) improved classifier robustness by emphasizing misclassified training samples during adversarial training. Their method led to a more robust classifier compared to traditional adversarial training approaches.

Moreover, Gui et al. (2019) presented Adversarially Trained Model Compression (ATMC), integrating adversarial training with model compression techniques such as pruning and quantization. ATMC enhances both the robustness and efficiency of deep neural networks by minimizing the worst-case adversarial loss while satisfying constraints on model sparsity and quantization precision.

Another approach is the incorporation of a local linearity regularizer (LLR) into adversarial training, as proposed by Qin et al. (2019). This regularizer has been proved in the work to be the upperbound of the loss function caused by adversarial examples. Their empirical analysis showed that networks trained with LLR exhibited improved robustness, especially on datasets like ImageNet.

Furthermore, Alayrac et al. (2019) introduced Unsupervised Adversarial Training (UAT) to effectively utilize unlabeled data for training robust classifiers. By combining supervised and unsupervised losses, UAT bridges the gap between natural and adversarial generalization, leading to improved adversarial robustness.

Additionally, Rade and Moosavi-Dezfooli (2021) proposed Helper-based Adversarial Training (HAT) to prevent excessive margin rise along initial adversarial directions. By introducing helper examples during training, HAT achieved improved performance on clean samples compared to traditional adversarial training methods.

Similarly, Zhang et al. (2021) introduced Geometry-Aware Instance-Reweighted Adversarial Training (GAIRAT), leveraging limited model capacity efficiently by fitting important data while ignoring unimportant ones based on their robustness against adversarial attacks and geometric distance from the decision boundary.

Moreover, Pang et al. (2022) proposed Self-COnsistent Robust Error (SCORE) to reconcile the trade-off between adversarial robustness and clean accuracy. SCORE encourages model predictions to align with the data distribution while maintaining robust optimization principles.

In addressing catastrophic overfitting, Jia et al. (2022) introduced FGSM-PGI, a prior-guided initialization strategy to mitigate it. By incorporating historical adversarial knowledge into the initialization process, FGSM-PGI effectively improved model robustness against adversarial attacks.

Dong et al. (2023) proposed Universal Inverse Adversarial Training (UIAT) to bridge the gap between adversarial examples and the high-likelihood region of their respective classes for robustness enhancement. UIAT leverages inverse adversarial examples to achieve this goal efficiently.

Moreover, He et al. (2023) introduced Self-Paced Adversarial Training (SPAT), a structured approach that explicitly guides the learning process from easy to complex adversarial examples, enhancing model robustness and generalization.

In exploring fairness in robustness, Ali Mousavi et al. (2023) introduced Fair Adversarial Retraining (FARMUR), aiming to achieve high robustness and fairness simultaneously by identifying vulnerable and robust data sub-partitions and applying fair adversarial retraining accordingly.

Additionally, Atsague et al. (2023) focused on minimizing both natural and adversarial risks through Penalized Modified Huber Regularization for Adversarial Training (PMHRAT), which strikes a balance between natural and adversarial accuracy while mitigating the impact of adversarial attacks on model performance.

Furthermore, Liu et al. (2023) proposed AdvMACER to enhance the robustness of randomized smoothed classifiers by maximizing the certified radius of adversarial examples. AdvMACER achieved significant improvements in model robustness and performance across various datasets.

Suzuki et al. (2023) introduced ARREST to mitigate the accuracy-robustness tradeoff in deep neural networks (DNNs) through adversarial finetuning, representation-guided knowledge distillation, and noisy replay. ARREST demonstrated improvements in both accuracy and robustness of trained DNNs.

Lastly, Cui et al. (2023) introduced the Improved Kullback-Leibler (IKL) Divergence loss to enhance stability and mitigate biases in adversarial training tasks. Their proposed approach effectively improved model robustness, particularly in adversarial training tasks.

Incorporating the latest addition, Park et al. (2024) proposed Adversarial Feature Alignment (AFA) as a new robust training method for deep neural network feature extractors. AFA effectively balanced robustness and accuracy by aligning features with the correct class manifold, achieving improvements in model performance. These approaches collectively contribute to advancing the field of adversarial robustness training by offering a diverse set of techniques to address the challenges posed by adversarial attacks.

2.2.2.3 Augmentation. The literature on augmentation-based adversarial robustness highlights diverse approaches and strategies employed to enhance robustness in machine learning models.

Rebuffi et al. (2021) proposed heuristics-driven augmentations as a strategy to mitigate robust overfitting without relying on external data. Their study found that techniques like Cutout or MixUp, when combined with early stopping, attenuated robust overfitting and led to a slower decline in robust accuracy compared to classical adversarial training. MixUp, in particular, demonstrated an ability to preserve robust

accuracy even at lower levels compared to other methods like Pad & Crop. Furthermore, the exploration of additional augmentation techniques such as CutMix revealed that it yielded higher "best" robust accuracy and was less prone to robust overfitting.

In contrast, Gowal et al. (2021) introduced an approach to enhance adversarial robustness through the utilization of low-quality generated data. By leveraging generated samples, particularly from a class-conditional Gaussian fit of the training data, they bolstered robustness in classification tasks. The study emphasized the importance of sample diversity and its complementarity with the original training set in enhancing robustness.

Moreover, Li and Spratling (2023) delved into understanding how the hardness and diversity of data augmentation influenced robust overfitting in adversarial training. Their findings led to the proposal of the Improved Diversity and Balanced Hardness (IDBH) augmentation scheme, aiming to mitigate robust overfitting by enhancing diversity and balancing hardness.

Similarly, Wang et al. (2023) focused on leveraging data generated by diffusion models, particularly the elucidating diffusion model (EDM), to enhance adversarial training without external datasets. This approach eliminated robust overfitting and reduced the generalization gap between clean and robust accuracy, presenting a promising direction for improving adversarial training effectiveness.

Furthermore, Yang et al. (2023) addressed defending against transfer-based black-box attacks through Data-centric Robust Learning (DRL). This method aimed to generate an augmented dataset for training models robust against adversarial examples, combining adversarial data augmentation and data selection strategies optimized for Cross-Entropy (CE) loss. Additionally, XDRL combined DRL with common techniques like synthetic data augmentation and alignment regularization loss functions to enhance robustness against black-box attacks.

These studies collectively contribute to advancing the understanding and effectiveness of augmentation-based methods in bolstering adversarial robustness in machine learning models. They explore various techniques, from heuristics-driven augmentations to the utilization of generated data and defense against black-box attacks, offering insights into the delicate balance between accuracy and robustness in adversarial training.

2.2.2.4 Other Techniques. Certain pieces of literature do not fit neatly within the previously outlined techniques, hence have been grouped in this section for an extended discussion and analysis. The work by Rice et al. (2020) explored the efficacy of early stopping to mitigate robust overfitting in adversarial training, showing that it can maintain optimal robust performance and prevent degradation. Similarly, Wang et al. (2023) explored approaches to address the Label-Feature Distribution Mismatch problem by deploying new adversarial training techniques, involving regularizing the distribution of adversarial examples to enhance robustness.

In a novel strategy to defend against adversarial attacks, Amich and Eshete (2021) presented Morphence, a method that transforms models into moving targets. By continuously renewing a pool of student models based on prediction confidence, it diversifies model predictions and enhances overall robustness in dynamic environments. Complementing this dynamic approach, Doan et al. (2022) combined adversarial training with Bayesian inference to represent uncertainty in parameters more accurately, aiming for a universal defense strategy across various learning contexts.

Several researchers focused on optimizing adversarial training to further push its limits in enhancing machine learning security. Yang et al. (2023) proposed novel optimization strategies that unify global and pairwise perturbation approaches, employing SVRG-based algorithms to manage computational difficulties. In contrast, Nuhu et al. (2023) introduced a two-phase method involving the DeepRobust

framework and strategic adversarial training to specifically enhance model resilience by distinguishing between weak and robust samples.

Innovating beyond traditional frameworks, Guan et al. (2023) suggested conducting training in the logit space rather than the input space using Endogenous Adversarial Examples (EAEs), aiming to enhance robustness while potentially streamlining the training process. In parallel, Gong et al. (2023) looked at improving training efficiency via gradient approximation techniques, employing partial Taylor series to approximate adversarial losses and simplify the training process.

Tao (2023) introduced the Meta-Adv framework to incorporate adversarial training within a meta-learning architecture, facilitating rapid adaption to new tasks while bolstering adversarial robustness. Similarly, Wang et al. (2023) suggested Robust Mode Connectivity (RMC) to minimize the maximum loss across various perturbations and select robust models along the connectivity path.

Focused on exploring the detailed dynamics of model robustness, Zhang et al. (2023) developed the ADVMOE framework, targeting enhanced robustness of MoE-CNN by considering interactions between router and pathway resilience. From a technical viewpoint, Kanai et al. (2023) tackled the nonsmooth nature of adversarial losses by introducing the Ensemble Stochastic Gradient Descent (EnSGD), promising robust training for neural networks under adversarial conditions.

Finally, Khan et al. (2023) have been advancing multi-prototype adversarial training by incorporating metric learning, illustrating a principled method to bolster performance and resilience of deep learning models in challenging applications.

Together, these approaches substantially contribute to developing a more robust and secure machine learning ecosystem, paving the way for the safe deployment of AI systems in varied real-world scenarios.

2.2.3 Architecture-Based Techinque

This section discusses the various studies and methodologies that have focused on adjusting the architectures of machine learning models to enhance their ability to withstand adversarial attacks. By exploring the modifications made to the structural design of these models, we gain insights into the innovative approaches and techniques employed to bolster their robustness in the face of potential adversarial challenges. This detailed examination allows us to better understand the nuanced strategies utilized in adapting the architectures of machine learning models to prioritize and enhance adversarial robustness, ultimately contributing to the broader discussion on the resilience of these models in real-world applications.

Rakin et al. (2018) proposed a novel approach to enhance adversarial robustness by quantizing activation functions within classifiers. First, they showed the fixed quantization that quantized an activation function into a number of bits. For example, they quantized a *sigmoid* function with this:

$$\frac{1}{2^n - 1} \times \text{round}[(2^n - 1) \times y],$$

where n is the number of bits, round$[\cdot]$ is a round function to make a float number of an integer, and y is the ouput of a *sigmoid* function. The work also described how to do the same thing for a *tanh* function. However, they did not mention how to quantize a ReLU function.

Next, the authors made the thresholds of the previous method tunable by an optimizer and called it the dynamic quantization. At the end, the performance and robustness of the dynamic quantization were better than the fixed quantization.

Qian and Wegman (2018) introduced techniques to construct L2-nonexpansive neural networks (L2NNNs), focusing on weight handling and activation function behavior to maintain crucial distance properties for robust performance. L2NNNs were based on the idea that the output of a layer should not expand greater than its

input as in:

$$y^T y \leq x^T x,$$

where x is an input and y is an output. From this inequality and $y = Wx$ where W was parameters of a layer, they derived it to

$$x^T (W^T W) x \leq x^T x$$

Therefore, minimizing $W^T W$ was a goal for L2NNNs to improve a model's robustness.

Meanwhile, Xie et al. (2019) addressed the vulnerability of convolutional networks to adversarial attacks by introducing feature denoising blocks, effectively mitigating perturbation amplification and improving adversarial robustness. These methods collectively highlight diverse strategies, ranging from activation quantization to feature denoising, aimed at bolstering adversarial robustness in neural networks.

Moreover, Sehwag et al. (2020) proposed a pruning approach based on empirical risk minimization (ERM) to make compact networks robust, identifying robust sub-networks while maintaining targeted accuracy metrics. Contrary to conventional expectations, Wu et al. (2021) found that wider models exhibited worse robust regularization effects, emphasizing the importance of considering perturbation stability in assessing robustness, thus linking model width to robustness. Additionally, Chen et al. (2022) outlined a novel approach to improving adversarial robustness through the integration of dropout techniques with the drift-diffusion model (DDM), enhancing model resilience against adversarial attacks through stochasticity.

In the domain of neural network architecture, Sooksatra et al. (2021) explored the enhancement of classical CNN performance through Quanvolutional Neural Networks (QNNs), demonstrating a novel approach to improving CNN robustness. Bubeck et al. (2021) examined the relationship between model size and robustness, proposing conjectures regarding neural network interpolation and robustness, while Bubeck and Sellke (2021) analyzed the probability of fixed functions providing

good approximate fits to random data, reaffirming the law of robustness. These studies collectively shed light on fundamental properties of neural networks and their robustness.

Furthermore, Zhang et al. (2021) and Zhang et al. (2022) presented innovative strategies for enhancing neural network robustness against adversarial attacks in the L_∞-norm domain, emphasizing the construction of inherently robust operations. Huang et al. (2021) explored the relationship between DNN architectural configuration, Lipschitzness, and adversarial robustness, providing insights into optimizing network architectures for enhanced robustness. Meanwhile, Peng et al. (2023) compared CNNs and Transformers to explore robust architectural components, offering specific design choices to enhance adversarial robustness.

Sooksatra et al. (2023) investigated the efficacy of capped ReLU functions in managing the amplification of adversarial perturbations, introducing caps on ReLU functions to control perturbation growth. Meanwhile, Huang et al. (2023) decomposed and studied the architectural design of adversarially robust residual networks, providing insights into designing robust residual blocks and optimizing network architectures for enhanced adversarial robustness. These studies further enrich the understanding of techniques aimed at improving neural network resilience against adversarial attacks.

Finally, Weitzner and Giryes (2023) analyzed the vulnerability of sparse coding algorithms to adversarial perturbations, drawing parallels with feature pruning in neural networks, while Lukasik et al. (2023) explored frequency domain analysis in CNNs, offering insights into utilizing frequency information for enhanced adversarial robustness. Additionally, Chitsaz et al. (2023) proposed Weight Clipping-Aware Training (WCAT) to mitigate quantization errors and enhance network robustness, and Yu et al. (2023) focused on universal adversarial patch attacks, proposing the Feature Norm Suppressing (FNS) approach to enhance CNN robustness. Finally, Ma et al. (2024) investigated the incorporation of random weights to enhance adversarial

robustness, while Wu et al. (2024) proposed the RSA algorithm for robustness enhancement by refining feature activations, suppressing redundant activations, and aligning feature spaces. These diverse methodologies collectively contribute to advancing the field of architecture-based adversarial robustness in neural networks.

2.2.4 Preprocessing-Based Techinque

This section explains a variety of methodologies and techniques focused on preprocessing images before inputting them into machine learning models, aimed at mitigating the presence of any potential adversarial perturbations. By carefully manipulating and refining the images prior to their utilization in the model, researchers and practitioners seek to minimize the risk of adversarial attacks by bolstering the model's ability to accurately interpret and classify visual data. This multifaceted process involves an array of preprocessing steps and strategies designed to enhance the robustness and resilience of the models, ultimately ensuring their efficacy and reliability in successfully handling image-based tasks. Through a comprehensive exploration of these preprocessing approaches, we gain valuable insights into the intricate methodologies employed to safeguard machine learning models against adversarial threats, contributing to advancements in the field of image recognition and classification.

Samangouei et al. (2018) introduced Defense-GAN, leveraging a Wasserstein Generative Adversarial Network (WGAN) trained on clean samples to denoise adversarial examples. This algorithm projected images onto the generator's range, minimizing reconstruction error to substantially reduce adversarial noise. Serving as a pre-processing step, Defense-GAN combated both white-box and black-box attacks without altering the classifier structure. Its non-linear nature and gradient descent loop during inference rendered it robust against gradient-based attacks. Interestingly, Defense-GAN theoretically maintained classifier performance without necessitating re-training if the GAN adequately represented the data.

Li et al. (2019) proposed the Certified Robust Classifier framework, ensuring certified robustness by bounding the tolerable size of attacks. Gaussian noise added to adversarial examples nullified perturbation effects before classification. The algorithm iteratively added Gaussian noise to pixels, estimating output distribution to calculate attack size bounds. Theoretical analyses demonstrated its certified robustness. Connecting robustness to adversarial and random noise, Stability Training with Noise (STN) improved adversarial robustness by enhancing robustness to random noise. Yang et al. (2019) introduced Matrix Extension Net (ME-Net), enhancing robustness by leveraging matrix estimation techniques. Random pixel masking destroyed adversarial structures, followed by ME-based reconstruction. ME-Net's efficacy in enhancing robustness was supported by empirical analyses revealing strong global structures inherent in images.

Bai et al. (2019) proposed PixelDefend, purifying images using PixelCNN, decomposing image likelihood into conditional distributions over pixels. The raster scan order limitation led to Hilbert-based PixelDefend (HPD), utilizing Hilbert curves to better capture pixel dependencies. HPD's theoretical guarantee of finding optimal clean images and ensemble version, EHPD, enhanced defense performance through pattern ensembling. Kabilan et al. (2021) introduced VectorDefense, employing Potrace vectorization to eliminate adversarial artifacts while retaining class-specific features. Binarization, despeckling, and smoothing components purified adversarial examples, enhancing classifier robustness.

Alfarra et al. (2022) added an anti-adversary layer to pretrained models, impeding adversaries by moving inputs away from decision boundaries. Unlike random perturbations, this layer boosted confidence in predictions without compromising clean accuracy. Mandal (2023) employed a U-shaped convolutional autoencoder to reconstruct original inputs from adversarial images, eliminating perturbations. GELU activation layers addressed the dying ReLU problem, facilitating effective learning.

Chen et al. (2023) explored distribution transfer for defense, utilizing diffusion models to enhance model performance on out-of-distribution samples.

Lyu et al. (2023) neutralized adversarial perturbations while preserving image structure using pixel masking and MAE reconstruction. Uniform masking and reconstruction schemes effectively mitigated adversarial noise, enhancing defense performance. Shah et al. (2024) proposed R-Blur to simulate human peripheral vision decline, incorporating biological principles into image processing for a more realistic representation.

2.2.5 Postprocessing-Based Techinque

This method involves manipulating the results produced by machine learning models to increase their resistance to adversarial attacks. It concentrates on improving post-processing procedures, executing remedial actions, or adding extra defensive layers to lessen the effects of adversarial threats. Various studies have utilized this strategy to enhance adversarial robustness. By carefully evaluating and enhancing the outputs, these methods aim to strengthen the models against potential weaknesses, ensuring they continue to perform well and reliably, even when faced with adversarial disturbances. The objective is to create more robust models that can endure complex attacks, thus boosting the overall security and durability of machine learning systems.

Papernot et al. (2016) introduced defensive distillation to mitigate vulnerabilities in deep neural networks (DNNs) against adversarial samples, employing distillation to enhance DNN robustness. The adaptation of distillation involved training two networks with the same architecture but different labels, leveraging additional knowledge encoded in probability vectors to improve generalization and resilience to adversarial perturbations.

Cohen et al. (2019) proposed randomized smoothing for constructing a smoothed classifier from a base classifier, providing robustness guarantees within a certain radius around the input against adversarial attacks. Meanwhile, Kang et al. (2021) proposed

the SODEF model architecture for classification tasks, incorporating a feature extractor, a neural ODE layer, and a fully connected (FC) layer to enforce stability against perturbations.

Additionally, Liu et al. (2023) introduced EsbRs to improve model robustness through mixed-model ensembles, showcasing advancements in ensemble studies. Furthermore, Vorácek and Hein (2023) utilized randomized smoothing to enhance L_1-certified robustness in binary and multiclass classification tasks, extending the method to incorporate box constraints for tighter upper bounds on minimal possible overlap.

Finally, Bai et al. (2023) proposed adaptive smoothing to reconcile the trade-off between accuracy and robustness in neural classifiers, achieving interpretable adjustment between the two at inference time, while Bai et al. (2024) introduced MixedNUTS, a training-free method optimizing the accuracy-robustness trade-off through nonlinear transformations of logits. These approaches collectively contribute to advancing the field of adversarial robustness in neural networks through post-processing step.

2.2.6 Combination-Based Techinque

Several research efforts have explored the integration of the aforementioned techniques to bolster adversarial robustness in machine learning models. By combining different strategies, these works aim to create a synergistic effect that enhances the overall defense mechanism against adversarial attacks. For instance, some works have combined defensive distillation with input preprocessing techniques to further harden the model's resistance to adversarial inputs. These comprehensive approaches not only address specific vulnerabilities but also provide a more holistic defense framework. By uniting various methodologies, researchers have been able to achieve significant improvements in the robustness and reliability of machine learning models, ensuring they can better withstand the evolving nature of adversarial threats.

Xu et al. (2017) proposed feature squeezing as a method to mitigate adversarial attacks on image classification models by reducing color depth and employing spatial smoothing. This approach aimed to decrease pixel variability and enhance classifier robustness, as demonstrated through evaluations on datasets like MNIST, CIFAR-10, and ImageNet. Furthermore, the detection method employed in this approach, comparing model predictions on original and squeezed samples, shared a common goal with the defense mechanism proposed by Meng and Chen (2017). MagNet, introduced by Meng and Chen (2017), aimed to identify and mitigate adversarial examples using a detector and a reformer. This system leveraged diversity in defense mechanisms to enhance effectiveness, similar to the multi-method approach of feature squeezing.

Meanwhile, Naseer et al. (2020) aimed to combine adversarial training and input processing methods into a single framework for improved robustness and clean image accuracy. The Neural Representation Purifier (NRP) model introduced in this framework shared a similar goal with MagNet's reformer component, aiming to reconstruct inputs to resemble normal examples and thereby enhance classifier robustness. Additionally, the exploration of activation function properties and their impact on adversarial training by Xie et al. (2020) provided insights into how different training techniques could affect model robustness.

Further insights into adversarial training techniques were provided by Gowal et al. (2020), who systematically explored factors affecting adversarial robustness. Their findings shed light on the importance of various factors such as model capacity, activation function choice, and handling of unlabeled data. These insights were complementary to the exploration of activation function curvature's role in adversarial training effectiveness by Singla et al. (2021), which challenged previous assumptions about the necessity of smooth activations for regularization effects.

Moreover, Qian et al. (2022) introduced a mechanism to guide the search for more robust network architectures by analyzing feature distortion in adversarial

examples. This approach aligned with the goals of Naseer et al. (2020) and Gowal et al. (2020) in enhancing model robustness. Additionally, Dai et al. (2022) investigated the impact of activation function shape on robust accuracy, providing insights that could inform the design of more robust networks.

Furthermore, defense techniques such as Adversarial Training on Purification (AToP) proposed by Lin et al. (2023) and the use of random projection filters by Dong and Xu (2023) offered additional strategies to enhance model robustness. These techniques focused on destructing adversarial perturbations and preserving distances among data points to improve the network's ability to defend against attacks.

Finally, Iijima et al. (2024), proposing a random ensemble method, and Huang et al. (2024), proposing a denoising defense approach, addressed the challenge of achieving robustness against various adversarial examples while maintaining high classification accuracy on clean images. These works provided innovative solutions to this challenge, highlighting the importance of addressing adversarial vulnerabilities from multiple perspectives.

Table 2.1. The techniques used by combination-based works

Approach	Detect	Train	Arch	Pre	Post
(Xu et al., 2017)	✓			✓	
(Meng and Chen, 2017)	✓			✓	
(Naseer et al., 2020)		✓		✓	
(Xie et al., 2020)		✓	✓		
(Gowal et al., 2020)		✓	✓		
(Singla et al., 2021)		✓	✓		
(Qian et al., 2022)		✓	✓		
(Dai et al., 2022)		✓	✓		
(Lin et al., 2023)		✓		✓	
(Dong and Xu, 2023)		✓	✓		
(Iijima et al., 2024)			✓	✓	✓
(Huang et al., 2024)		✓		✓	

Table 2.1 provides a comprehensive overview of the techniques utilized in combination-based approaches for enhancing adversarial robustness. Each work employs different combinations of these techniques to achieve robust adversarial defenses.

Several notable patterns emerge from this data. Two works, Xu et al. (2017) and Meng and Chen (2017), utilized both detection and preprocessing techniques. This combination focuses on identifying adversarial examples before they reach the model and modifying the input data to mitigate adversarial effects. In contrast, works such as Naseer et al. (2020), Lin et al. (2023), and Huang et al. (2024) combined training techniques with preprocessing. This approach leverages robust training procedures and prepares data in a way that enhances model resilience.

A significant number of works paired training with architecture-based techniques. This combination involves modifying the model's structure and applying robust training methods to strengthen defenses against adversarial attacks. Uniquely, (Iijima et al., 2024) combined architecture-based, preprocessing, and postprocessing techniques. This comprehensive approach modifies the model architecture, prepares data for enhanced robustness, and applies post-processing methods to further refine model predictions and reduce adversarial impact.

Table 2.1 highlights that training techniques are the most commonly used method, appearing in 10 out of the 12 works. This trend suggests that training methods are viewed as foundational to improving adversarial robustness. Their frequent use may be due to their direct impact on model performance and the flexibility they offer in combination with other techniques. Architecture-based techniques are the second most common, used in 7 out of 12 works. This indicates a strong focus on modifying the internal structure of models to make them more resistant to adversarial attacks. The popularity of these techniques reflects the importance of building robustness directly into the model architecture.

Preprocessing methods are also widely used, appearing in 6 out of the 12 works. These methods are often combined with training and architecture-based techniques, underscoring the importance of preparing input data to withstand adversarial manipulations. In contrast, detection techniques are used in only 2 works, while postprocessing is used in just 1 work. This suggests that while detection is important, it may be less favored compared to more proactive approaches like training and preprocessing. Postprocessing, being the least used, might be considered less effective or more supplementary in nature.

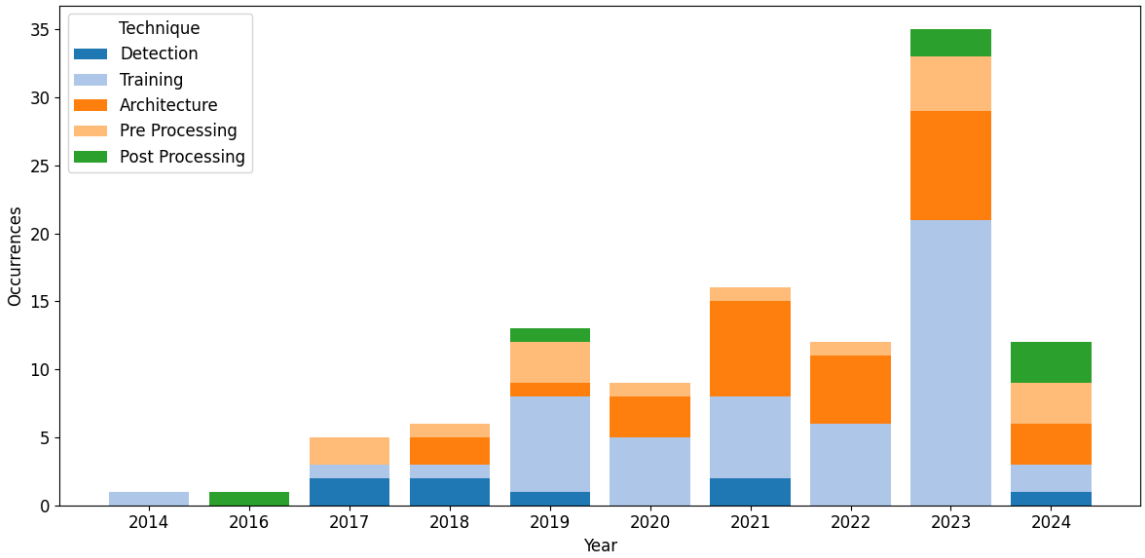
These trends and patterns imply several important directions for future research. The trend towards combining multiple techniques indicates a recognition of the complex nature of adversarial threats. Future research may continue to explore integrative approaches that combine the strengths of various methods to develop more comprehensive defenses. Additionally, the limited use of postprocessing and detection techniques suggests potential areas for innovation. Researchers might focus on developing more effective detection algorithms and postprocessing methods that can complement existing approaches.

As training techniques often increase computational costs, there is an implicit need for developing more efficient and scalable solutions. This could drive research towards optimizing existing techniques or creating novel methods that balance robustness with computational efficiency. The diverse combinations of techniques also imply potential benefits from interdisciplinary collaboration. Insights from fields such as computer vision, data preprocessing, and cybersecurity can be integrated to enhance the robustness of machine learning models.

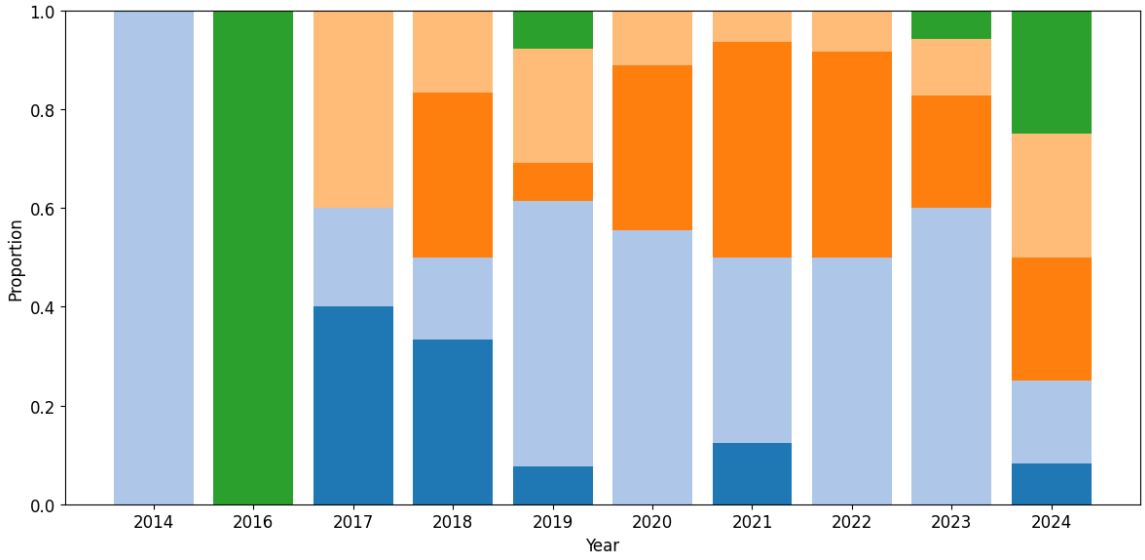
Table 2.1 not only highlights the prevalent techniques in combination-based adversarial robustness works but also points to important trends and future directions in the field. The dominance of training and architecture-based techniques, along with

the rising use of preprocessing methods, reflects the ongoing efforts to develop robust and resilient models against adversarial attacks.

2.2.7 Discussion



(a) Occurences



(b) Proportions

Figure 2.1. The occurrences and proportions of types of the approaches by each year in our literature review

We discussed the approaches for each technique in the previous sections. This section demonstrates the trend of the growth of works in adversarial robustness. Figure 2.1 illustrates the number of works and the number of works in each technique per year since 2014. As seen in the figure, the number of works in adversarial robustness has grown rapidly from 2014 to 2024, with a particularly notable increase in the use of training techniques. This technique has gained popularity because it is both highly effective and straightforward to implement. However, it is important to note that the training time for these techniques has significantly increased.

Consequently, the adoption of other techniques has also risen over the past years. For instance, Figure 2.1b presents the normalized version of Figure 2.1a, revealing a substantial proportion of works based on architecture-based and preprocessing-based techniques. This diversification in techniques suggests a response to the limitations and challenges posed by training-based methods, such as the increased computational costs.

Overall, these figures not only demonstrate the active and evolving nature of research in adversarial robustness but also imply several important trends and shifts in the field. The initial dominance of training techniques points to their early success and ease of application. However, as the field has matured, researchers have explored and increasingly adopted a wider array of methods, indicating a recognition of the need for more efficient or complementary approaches.

Furthermore, the rise in architecture-based and preprocessing-based techniques implies a broader understanding and approach to adversarial robustness. These methods can offer benefits such as reduced training times and potentially greater generalizability. The growing diversity in research approaches may also reflect an interdisciplinary collaboration, where insights from different domains are integrated to enhance the robustness of machine learning models.

The figures highlight not only the growth in the number of works but also the evolution and diversification of techniques in adversarial robustness. This trend suggests that the research community is actively seeking more effective and efficient ways to combat adversarial attacks, leading to a more comprehensive and nuanced understanding of the field.

2.3 Datasets

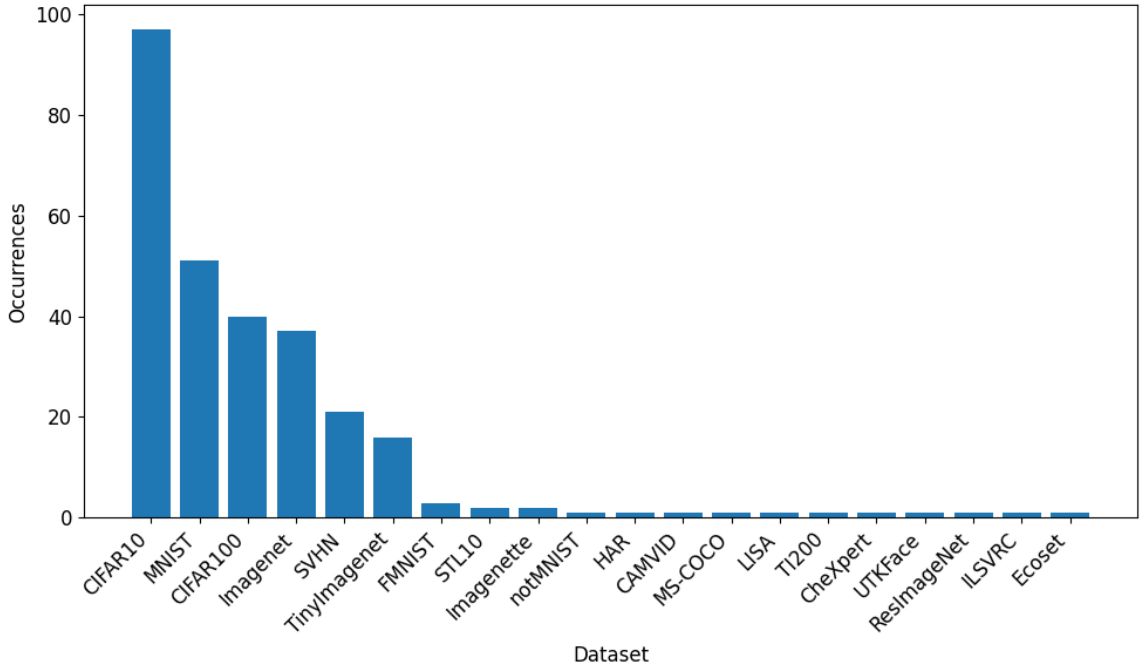
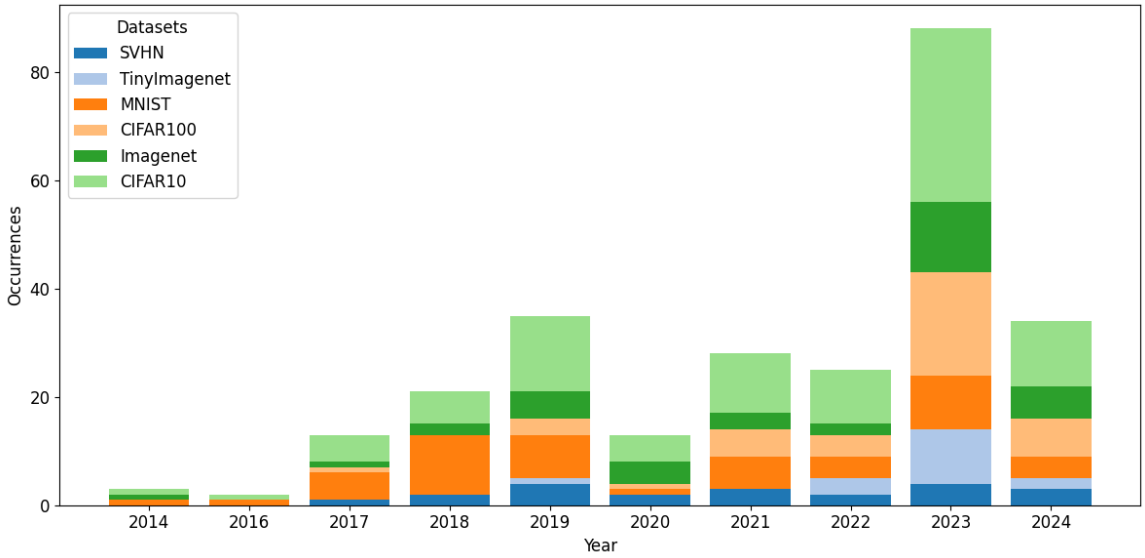
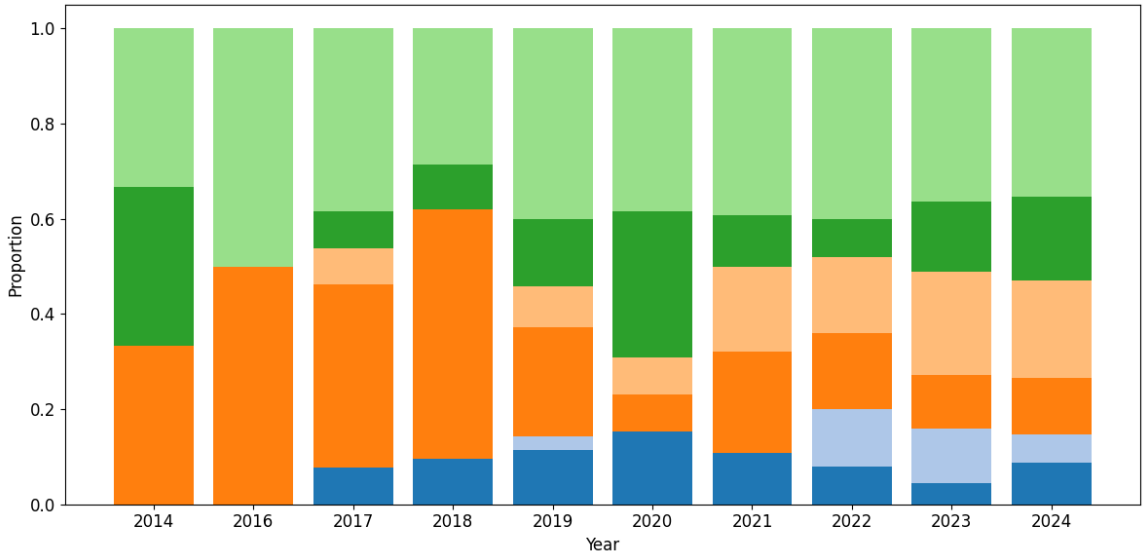


Figure 2.2. The occurrences of datasets over the approaches in our literature review

This section discusses the datasets used in adversarial robustness approaches. Figure 2.2 presents the frequency occurrences used in these works. The most frequently utilized datasets include CIFAR10 (Krizhevsky et al., 2009), MNIST (Deng, 2012), CIFAR100 (Krizhevsky et al., 2009), Imagenet (Russakovsky et al., 2015), SVHN (Netzer et al., 2011), TinyImagenet (Chrabaszcz et al., 2017), and FMNIST (Xiao et al., 2017a).



(a) Occurences



(b) Proportions

Figure 2.3. The occurrences and proportions of datasets over the approaches by each year in our literature review

Figure 2.3 shows the frequency of the top 6 most frequent-used datasets utilized in the literature. Between 2014 and 2018, MNIST and CIFAR10 were particularly popular. These datasets are relatively small, making them suitable for early research in adversarial robustness, which often involves extensive training time. The simplicity

and size of these datasets allowed researchers to quickly test and iterate on their approaches without the computational burden of larger datasets.

After 2018, there was a noticeable shift towards using larger datasets such as CIFAR100, TinyImagenet, and Imagenet. This shift likely reflects the maturation of the field and the increased attention it has garnered from various institutions. As adversarial robustness research has progressed, there has been a greater emphasis on demonstrating the effectiveness of techniques on more complex and diverse datasets. This trend suggests that researchers are moving towards more realistic and challenging scenarios, ensuring that their methods can scale and perform well on large-scale data typically encountered in practical applications.

The implications of these trends are significant. Initially, the reliance on smaller datasets like MNIST and CIFAR10 enabled rapid advancements and the development of foundational techniques in adversarial robustness. These early stages were crucial for understanding the fundamental challenges and creating initial solutions. However, the eventual shift to larger datasets indicates a push towards more robust and generalizable methods that can handle the complexities of real-world data.

This evolution also implies a growing computational capability within the research community. The increased use of large datasets like Imagenet, which requires substantial computational resources, suggests that institutions are investing more in this area, providing the necessary infrastructure to support such resource-intensive research. Additionally, it reflects a confidence in the scalability of new techniques and their applicability to more demanding tasks.

Moreover, the diversity in dataset use highlights the importance of evaluating adversarial robustness across various types of data. By incorporating datasets with different characteristics—such as the digit-focused MNIST, the object-centric CIFAR series, and the complex, high-resolution images of Imagenet—researchers can ensure that their methods are versatile and effective across a wide range of scenarios. This

comprehensive evaluation is crucial for developing universally robust models that can withstand adversarial attacks in different contexts.

The data also implies a possible future trend where new datasets might emerge as benchmarks for adversarial robustness, driven by evolving research needs and technological advancements. As the field continues to grow, the diversity and complexity of datasets will likely expand, fostering the development of even more sophisticated and resilient models.

The figures illustrate not only the growth in the number of works but also the evolution and diversification of datasets used in adversarial robustness research. This trend towards larger and more varied datasets implies a maturing field focused on developing more generalizable and scalable solutions, backed by increasing computational resources and institutional support.

2.4 Conclusion

Building upon the insightful analysis presented in the earlier sections of this document, it is clear that architectural-based techniques dealing with adversarial robustness have seen a noticeable growth over the years. Despite this evident advancement, there appears to be a notable gap in the experimental exploration of capping activation functions, such as ReLU functions. To the best of our knowledge, our research is pioneering in this area by being the first to rigorously experiment with and investigate the potential and implications of applying caps to such activation functions within machine learning models. This venture into uncharted territory is intended to provide insights that could drive further enhancements in model robustness against adversarial attacks.

Furthermore, our comparative analysis of various datasets utilized in the field of adversarial machine learning reveals a strong preference towards certain datasets. Among these, the MNIST, CIFAR10, CIFAR100, TinyImagenet, and Imagenet datasets stand out as particularly popular due to their diverse and challenging nature, which

makes them suitable for testing the robustness of machine learning models under adversarial conditions. Our experimental design, therefore, includes these well-regarded datasets to ensure that our findings are relevant and comparable to existing studies. However, we have decided to exclude the Imagenet dataset from our experiments. The decision stems from practical constraints; notably, the sheer size of the Imagenet dataset poses significant resource challenges. It requires an extensive amount of computational power and time for training and testing models, which are beyond our current capabilities.

Incorporating these popular datasets, barring Imagenet, allows us to engage with the broader discourse in this area while managing our resources effectively. By focusing on architectures and activation functions, particularly exploring the novel area of capping functions like ReLU, our work aims to contribute valuable knowledge to the field of adversarial robustness. Through this approach, we hope to uncover findings that could spur further research and innovation, particularly in developing more secure and resilient machine learning systems that can withstand increasingly sophisticated adversarial attacks.

CHAPTER THREE

Problem of ReLU Activation Functions

This chapter has been published as: Sooksatra, Korn, et al. "Is relu adversarially robust?" LXAI Workshop at the Fortieth International Conference on Machine Learning, 2023. https://doi.org/10.52591/lxai202307232

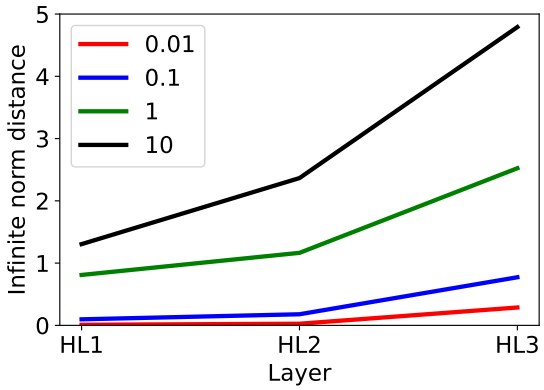
ReLU activation functions have been widely used in deep-learning models due to their ability to accelerate the training process and address the vanishing gradient problem. Unlike Sigmoid and Tanh activation functions, ReLU activation functions have many spaces for gradient computation, making them more friendly to backpropagation. However, this property that makes ReLU functions worthwhile makes them weak in deep-learning models regarding adversarial examples. Given that ReLU functions allow many tiny perturbations in inputs to be enlarged over the hidden layers by the operations in the models, these tiny perturbations can result in a significant difference in the output layer, making the model vulnerable to adversarial examples.

3.1 Enlarged Perturbations

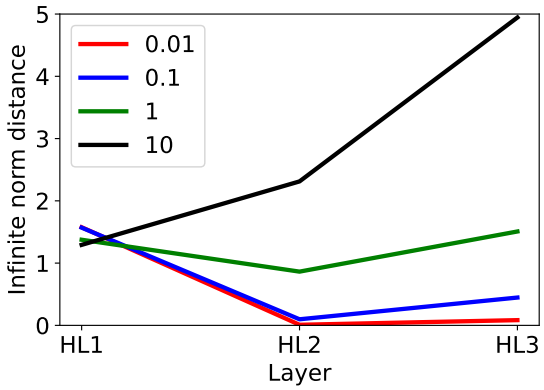
This section[1] shows the benefit of capping the ReLU functions. Inspired by ReLU6 in (Sandler et al., 2018), we found that capping ReLU activation functions can stop the perturbations from growing over the layers in our subsequent experiments that show the growing perturbations in the hidden layers for various max values. We use the MNIST dataset (LeCun et al., 2010) and train its classifier consisting of three dense hidden layers whose sizes are 392, 196, and 98. After that, we utilize Projected Gradient Descent (PGD) attack on the classifier and the test dataset with the perturbation bound of 20/256, step size of 2/256 and the max iteration of 20.

[1]The main difference between the published version and this chapter is that, here, we rewrite the first two sentences of the first paragraph and separate the figure used in this section into two figures to improve readability.

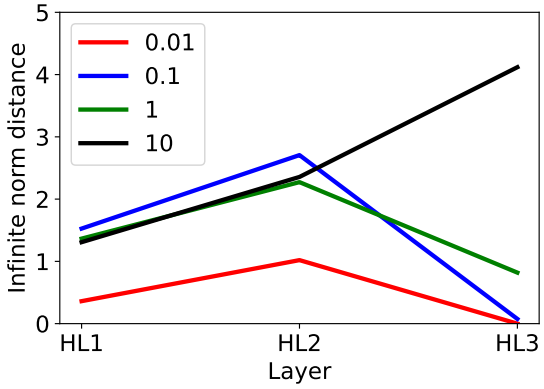
Then, we obtain adversarial examples. Next, we train other classifiers and cap different hidden layers (i.e., the first hidden layer (HL1), the second hidden layer (HL2), the third hidden layer (HL3) and all the hidden layers (HL123)). Also, we cap them with diverse max values (i.e., 0.01, 0.1, 1, 10, and 100). Figure 3.1 and 3.2 demonstrate that ReLU functions with high max values allow perturbations to become huge over the layers. On the other hand, capping them with low values can mitigate such an effect. Further, it is intuitive that the difference significantly goes down at the layer capped, as seen in the figure.



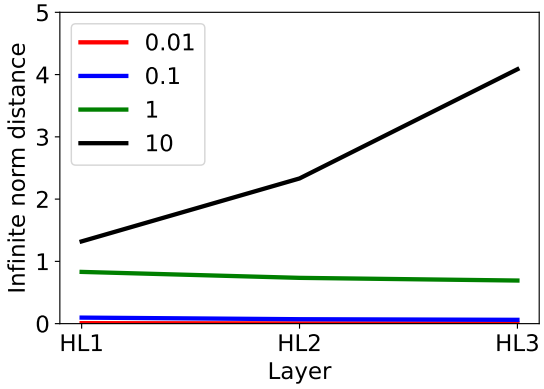
(a) Cap the first hidden layer with the infinite norm distance.



(b) Cap the second hidden layer with the infinite norm distance.

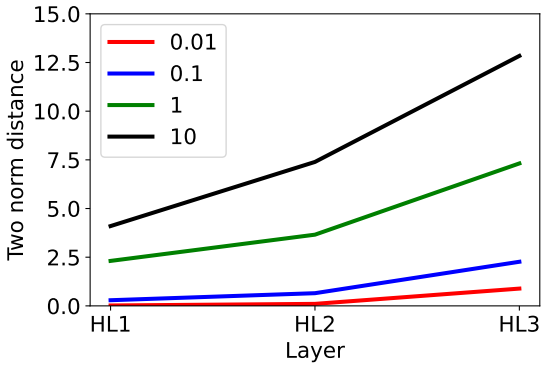


(c) Cap the third hidden layer with the infinite norm distance.

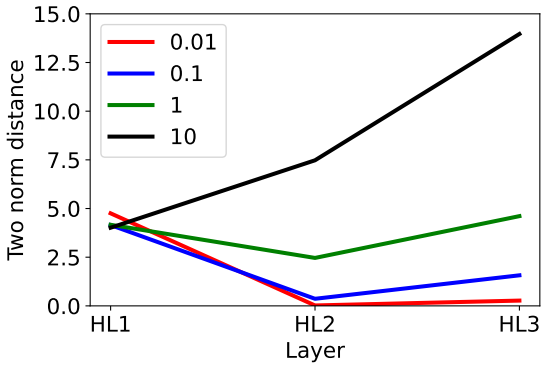


(d) Cap all the hidden layers with the infinite norm distance.

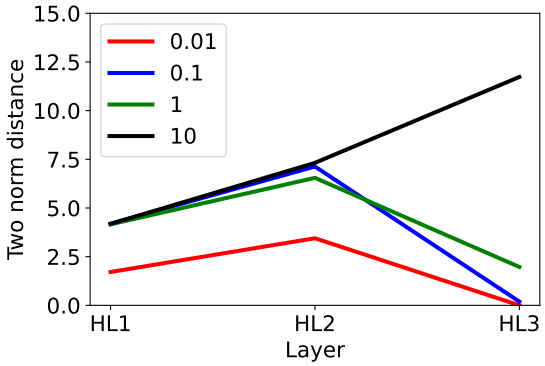
Figure 3.1. The L_∞ distance between each hidden layer's outputs resulted from passing clean samples and adversarial examples.



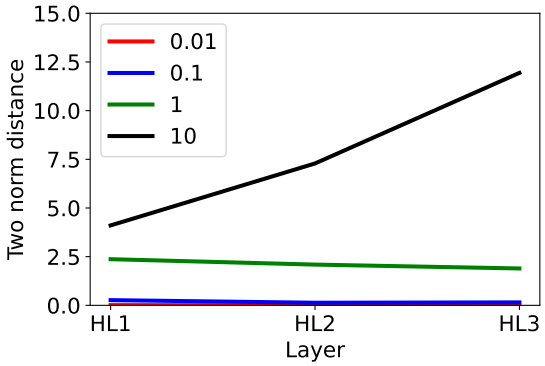
(a) Cap the first hidden layer with the two norm distance.



(b) Cap the second hidden layer with the two norm distance.



(c) Cap the third hidden layer with the two norm distance.



(d) Cap all the hidden layers with the two norm distance.

Figure 3.2. The L_2 distance between each hidden layer's outputs resulted from passing clean samples and adversarial examples.

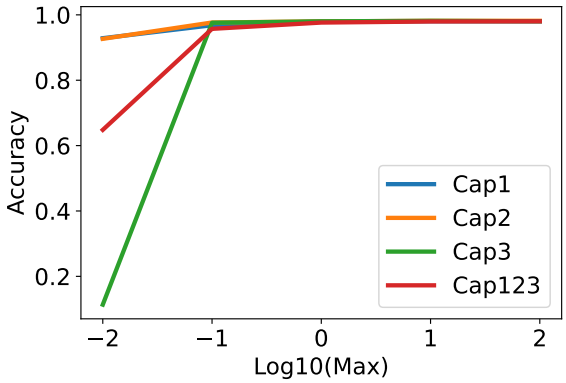


Figure 3.3. Accuracy achieved by classifiers with different capped hidden layers and max values on MNIST test dataset.

Although capping ReLU functions reduces the growth of perturbed values that may significantly alter the output, we found that when we set the max value to be very

low, the classifier would underfit the dataset due to the vanishing gradient problem, as demonstrated in Figure 3.3. Also, capping all the hidden layers achieved a slightly lower performance than capping only one hidden layer, as seen in the figure when the max value is 0.1. Therefore, there is a tradeoff between the network's ability to be trained and its sensitivity to tiny perturbations in this phenomenon.

3.2 Capped ReLU Function

We established that capping ReLU functions could significantly reduce the perturbations over the layers in the previous section. Therefore, this section shows the formal definition of the capped ReLU function.

A capped ReLU function is a general ReLU function capped with a value. Hence, we can formulate this function as

$$\max(0, \min(z, m)), \tag{3.1}$$

where z is the function's input and m is a max value that caps the function. As seen in Figure 3.2a where m is the capping value (i.e., 0.01, 0.1, 1 and 10), reducing m can control the growing perturbations efficiently.

Sigmoid and Tanh activation functions can be good candidates for providing adversarial robustness since their values have the highest and lowest values. However, the output spaces of these functions are still too wide (i.e., from 0 to 1 for Sigmoid and from -1 to 1 for Tanh). As seen in Figure 3.1 and 3.2, the ReLU functions with the capping value of 1 cannot prevent the growth of perturbations over the hidden layers. Therefore, these functions are not robust enough against adversarial examples.

CHAPTER FOUR

Static-Max-ReLU Activation Functions

This chapter has been published as: Sooksatra, Korn, et al. "Is relu adversarially robust?" LXAI Workshop at the Fortieth International Conference on Machine Learning, 2023. https://doi.org/10.52591/lxai202307232

The previous chapter introduces a method to enhance adversarial robustness in machine learning models by capping ReLU functions with maximum values, thus mitigating the impact of enlarged perturbations. 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. Theoretical analyses will be presented to demonstrate the enhanced robustness of this proposed function compared to a general ReLU function. Furthermore, empirical experiments will be detailed in the subsequent sections to validate its improved robustness empirically.

4.1 Theorectical 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.[1]

Theorem 4.1.1. The outputs of S-ReLU functions always have fewer perturbations than or equal perturbations to the outputs of ReLU functions, given the same inputs.

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 $\text{act}(\cdot)$)

[1]The material in this section does not appear in any of our current publications.

is

$$o_j^l = \text{act}\left(\sum_i w_{ij}^{l-1} \cdot o_i^{l-1}\right). \tag{4.1}$$

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

$$\begin{aligned} o_j^{l^*} &= \text{act}\left(\sum_i w_{ij}^{l-1} \cdot (o_i^{l-1} + \delta_i^{l-1})\right) \\ &= \text{act}\left(\underbrace{\sum_i w_{ij}^{l-1} \cdot o_i^{l-1}}_{A} + \underbrace{\sum_i w_{ij}^{l-1} \cdot \delta_i^{l-1}}_{B}\right), \end{aligned} \tag{4.2}$$

where o^* means that the output has 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. There are six cases that 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 ReLU because $B > m - A$ from 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 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 ReLU because one of the conditions is $A > m$. Then, $|m| < |A|$.
- Case 6: $A \leq m$ and $A + B \leq m \rightarrow |o_j^l - o_j^{l^*}|$ of both ReLU and S-ReLU are the same because S-ReLU behaves the same as ReLU.

These results are summarized in Table 4.1 and show that the output of S-ReLU will never exceed the one of ReLU. Then, the theorem is valid. □

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 4.1.1, 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

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

Table 4.1. 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	Outputs' 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	

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.

Theorem 4.1.2. When the max value (m) of S-ReLU in a layer reduces, the layer's outputs between clean samples and adversarial examples are closer.

Proof. This theorem can be easily proved by the information in Table 4.1 summarized from the proof of Theorem 4.1.1. When m decreases, S-ReLU's $|o_j^l - o_j^{l^*}|$ also decreases or remains the same. □

According to Theorem 4.1.2, we can reduce the max values of S-ReLU to reduce the Lipschitz constant and eventually improve more robustness. However, this technique may harm the overall performance if the max values are too low.

4.2 Effect of Capped Layer's Size on Robustness

In this section, we explore the question of whether the width of the layer (narrow or wide) containing the capped neurons affects robustness. Therefore, we conducted the following experiment.

4.2.1 Experimental Explanation and Setting

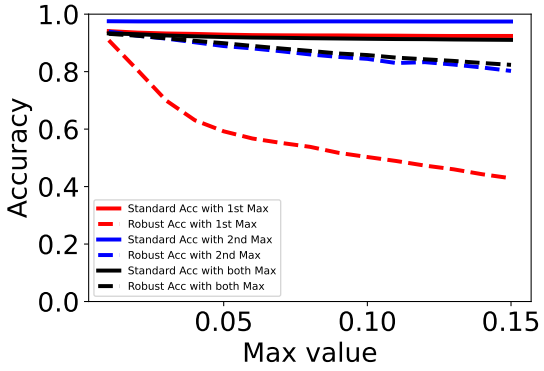
First, we trained a classifier with some layers capped with an initial max value to control the value after the ReLU function. Then, we evaluated this classifier in terms of accuracy on clean test samples (i.e., standard accuracy) and adversarial examples (i.e., robust accuracy) and the success rate of an attack. However, we cannot

rely only on the initial max value because it may not make the classifier the most robust. Therefore, we evaluate this classifier with several max values (i.e., from 0.01 to 0.15 in our experiments) to determine the max value that promotes robustness and does not sacrifice much standard accuracy. We used the MNIST dataset for this experiment. Also, we created two kinds of classifiers: a "general" two-hidden-layer dense network and a "reversed" two-hidden-layer dense network. The former consists of an input layer, a 392-neuron layer with ReLU activation, a 196-neuron layer with ReLU activation, and the output layer with Softmax activation. In the latter, only the hidden layers are swapped. Further, the attack we used for this experiment is Projected Gradient Descent (PGD) (Madry et al., 2017) because it is one of the strong attacks and is widely used for adversarial robustness evaluation.

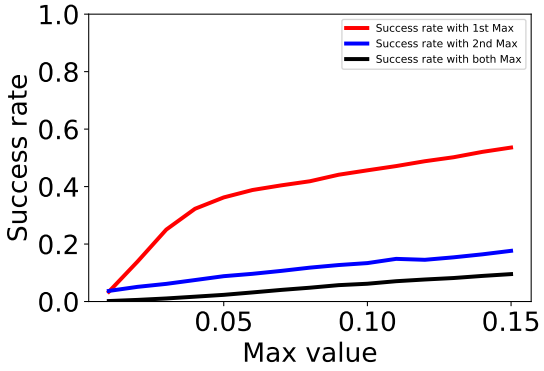
4.2.2 Results

Figure 4.1 shows the result of this experiment with the general network. It demonstrates that with the initial max values of 0.01 and 0.1, capping the second hidden layer surprisingly outperforms the first hidden layer and capping both the hidden layers. However, with the initial max value of 1, capping the second hidden layer underperforms the others when the max value is greater than 0.05 in terms of robustness. Nonetheless, the robustness of the classifier being capped at the second hidden layer with a max value less than 0.05 is higher than the others in all the max values. Therefore, capping the second hidden layer is the best solution for this general network. Because the second hidden layer has lower neurons than the first hidden layer, capping the small layer is better than capping the large layer. However, this result is derived from a specific network. Next, we experiment with the reversed network.

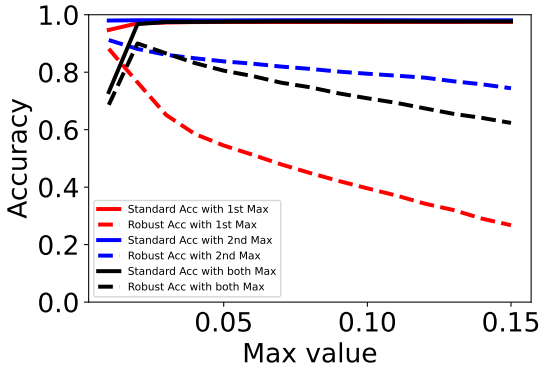
Figure 4.2 shows the results of the same experiment from the reversed network. Capping both the hidden layers outperforms the others in most cases. Also, capping the first hidden layer outperforms capping the second hidden layer in most cases. Since



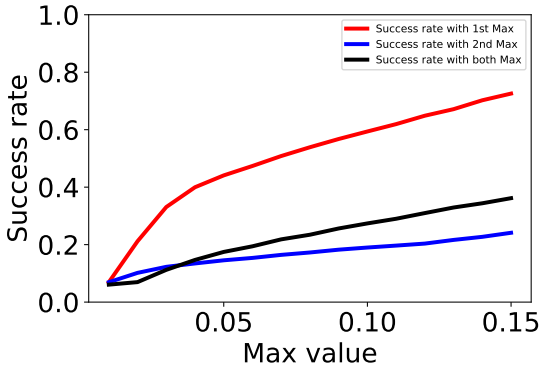
(a) Initial max = 0.01 with accuracy score.



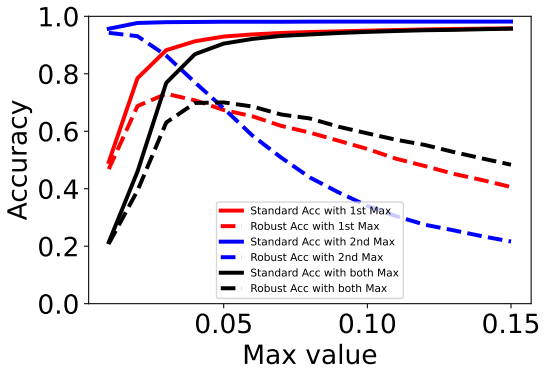
(b) Initial max = 0.01 with success rate.



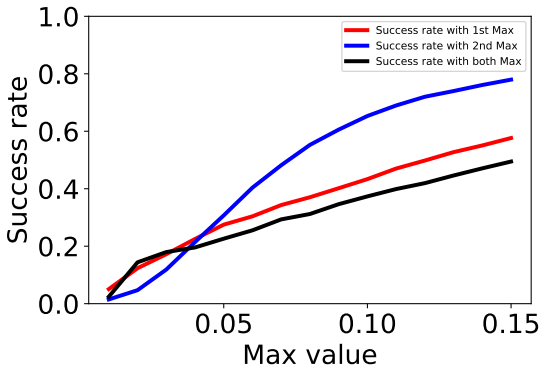
(c) Initial max = 0.1 with accuracy score.



(d) Initial max = 0.1 with success rate.



(e) Initial max = 1 with accuracy score.



(f) Initial max = 1 with success rate.

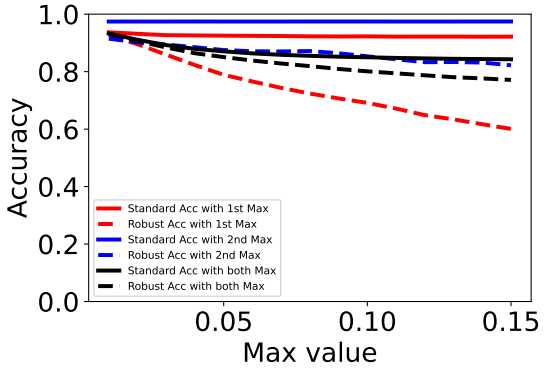
Figure 4.1. Standard accuracy, robust accuracy, and success rate of a two-hidden-layer classifier under a PGD attack across various maximum perturbation values. Standard accuracy refers to the classifier's performance on clean samples, robust accuracy indicates its performance on adversarial examples, and success rate is the proportion of correctly classified clean samples that the attack successfully converts into adversarial examples..

the first hidden layer contains fewer neurons than the second hidden layer, capping a small layer is better than capping a large layer in this case. We can summarize that capping a bottleneck layer would result in the most robustness.

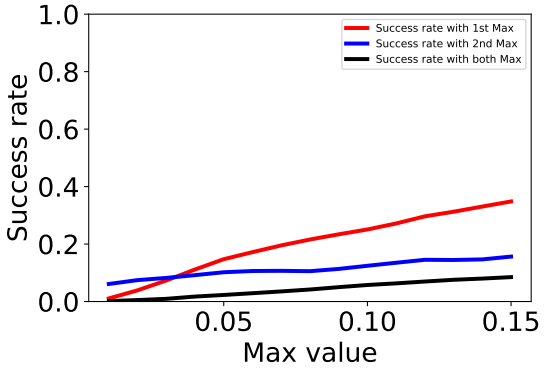
4.3 Effect of Capped Layer's Order on Robustness

In this section, we discuss the question of whether capping an early or deep layer in a classifier can provide the most robustness. We conducted the same experiment as in the previous section. However, we built another classifier consisting of the input layer, two 784-neuron hidden layers with ReLU activations, and the output layer with Softmax activation. Noticeably, the hidden layers' sizes are equal to see which layer affects the most in terms of robustness.

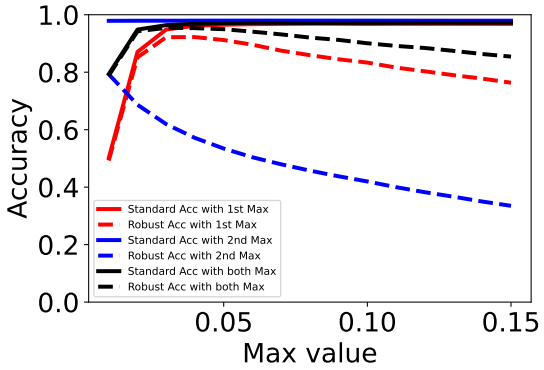
Figure 4.3 shows the results of this experiment. The classifier capped at the first hidden layer performs relatively better when the initial max value grows. Evidently, at the initial max value of 0.01, capping the second hidden layer is better than the first hidden layer. This phenomenon is intuitive because if we cap only the first hidden layer, an adversary can eventually find a way to amplify adversarial perturbations in the subsequent layers. On the other hand, capping at the second hidden layer can effectively prevent this from occurring. However, increasing the initial max value results in the opposite consequence. Therefore, capping the deep layer is recommended with a very low initial max value, and the early layer is preferred with a medium to high initial max value.



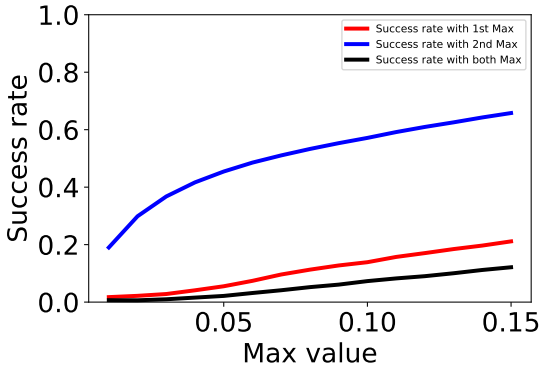
(a) Initial max = 0.01 with accuracy score.



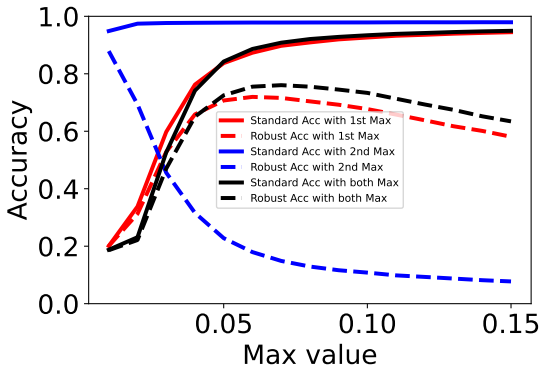
(b) Initial max = 0.01 with success rate.



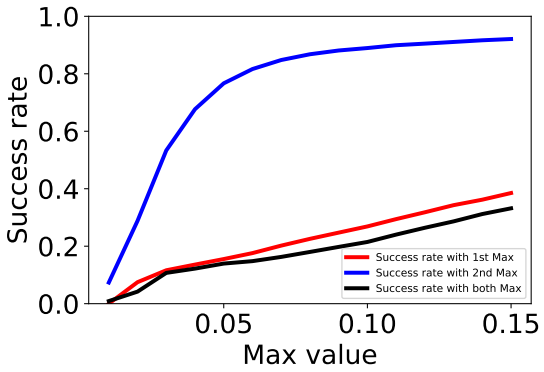
(c) Initial max = 0.1 with accuracy score.



(d) Initial max = 0.1 with success rate.

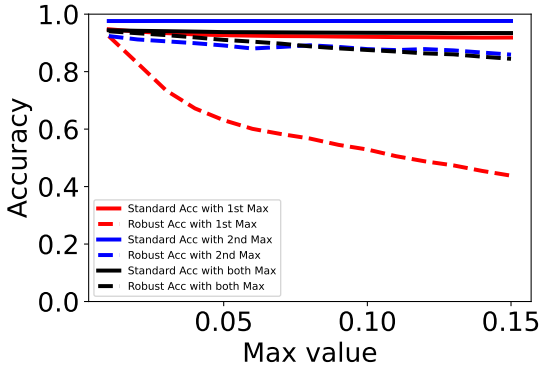


(e) Initial max = 1 with accuracy score.

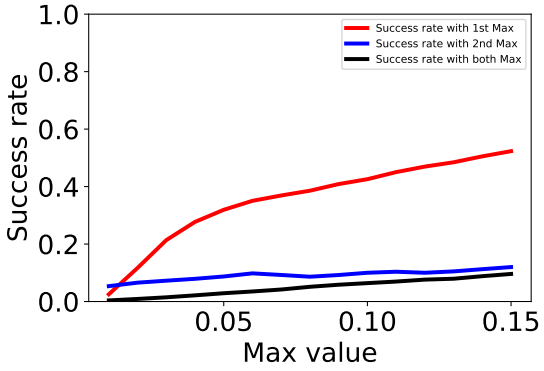


(f) Initial max = 1 with success rate.

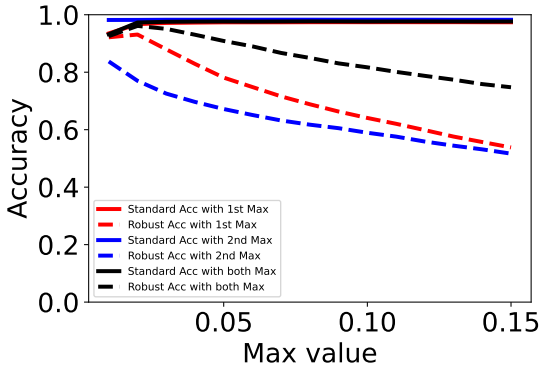
Figure 4.2. Standard accuracy, robust accuracy, and success rate of a reversed two-hidden-layer classifier under a PGD attack across various maximum perturbation values. Standard accuracy refers to the classifier's performance on clean samples, robust accuracy indicates its performance on adversarial examples, and success rate is the proportion of correctly classified clean samples that the attack successfully converts into adversarial examples.



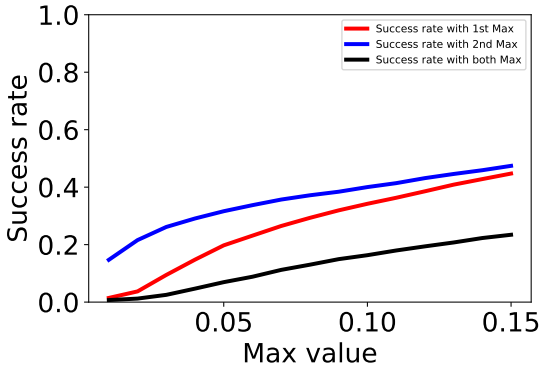
(a) Initial max = 0.01 with accuracy score.



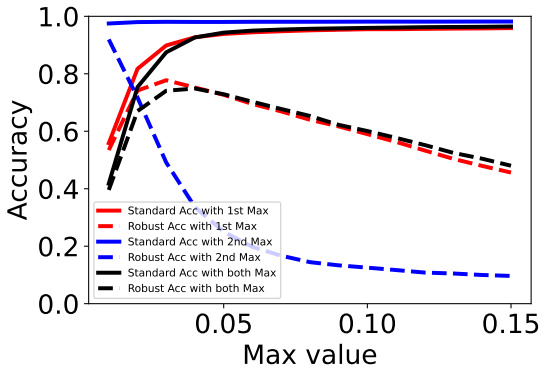
(b) Initial max = 0.01 with success rate.



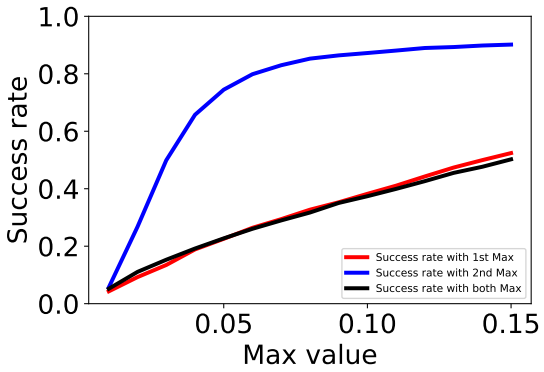
(c) Initial max = 0.1 with accuracy score.



(d) Initial max = 0.1 with success rate.



(e) Initial max = 1 with accuracy score.



(f) Initial max = 1 with success rate.

Figure 4.3. Standard accuracy, robust accuracy, and success rate of a equal two-hidden-layer classifier under a PGD attack across various maximum perturbation values. Standard accuracy refers to the classifier's performance on clean samples, robust accuracy indicates its performance on adversarial examples, and success rate is the proportion of correctly classified clean samples that the attack successfully converts into adversarial examples.

4.4 Zero Gradient Experiment

In this experiment[2], we aim to provide further empirical evidence to support the validity of our previous results. To this end, we have modified the projected gradient descent (PGD) attack method to include a new stopping criterion, which we refer to as "zero gradients". Specifically, instead of terminating the attack once an adversarial example has been found, we continue the attack until the gradient of the objective function is zero. This modification allows us to assess the robustness of a targeted classifier in a more meaningful way.

We assume that a classifier is robust if the location where the zero gradients are found is close to the original, clean input sample. This is because if an attacker encounters a zero gradient, they will no longer be able to perform any gradient-based attacks, such as the fast gradient sign method (FGSM) or PGD. To measure the distance between the clean sample and the location where the zero gradients are found, we use the Euclidean distance.

However, it should be noted that this method can fail if zero gradients are not found within the maximum number of iterations. Figure 4.4 illustrates examples of both successful and unsuccessful scenarios. To obtain a more comprehensive understanding of the robustness of the classifiers, we compute the average distance only from the test samples where zero gradients are found. We use the MNIST dataset, along with the classifiers from the previous sections for this experiment.

Figures 4.5, 4.6, and 4.7 show the results of the zero gradient experiment with the general, reversed and equal networks, respectively. Interestingly, we found that these results aligned with the previous experiments in Section 4.2 and 4.3. Therefore, we can conclude that the results in those sections are valid.

[2]We split the figure in this section into two figures to enlarge the fonts.

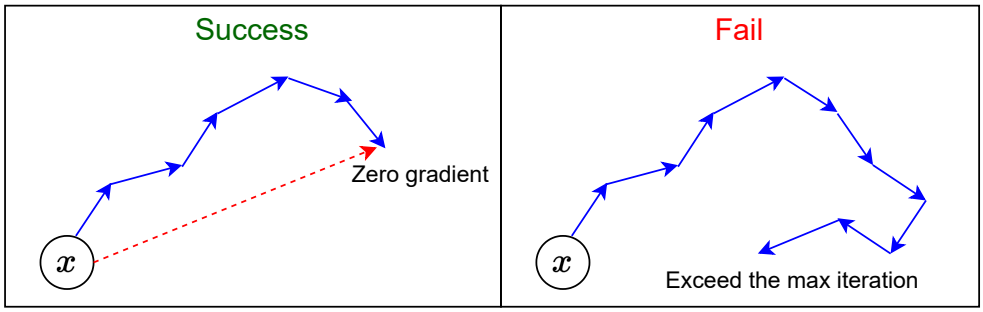
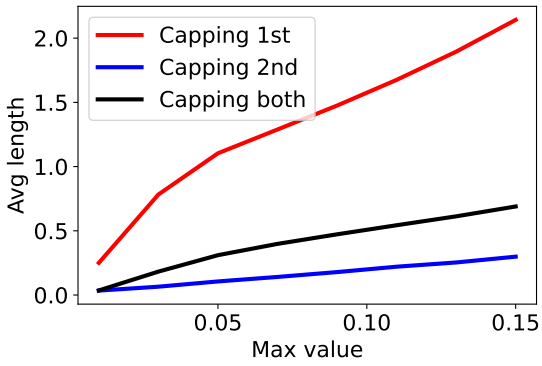
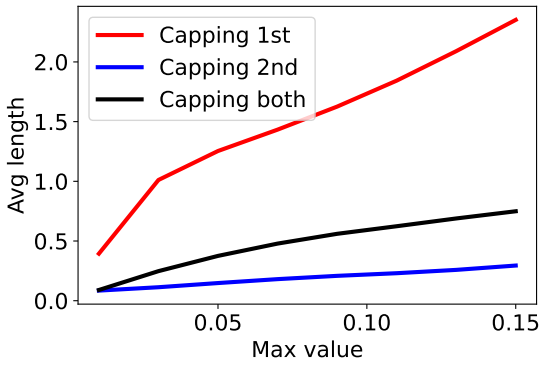


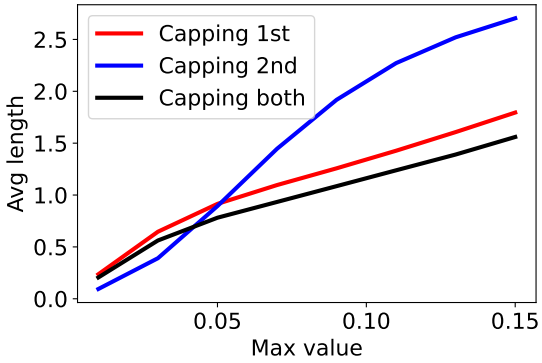
Figure 4.4. Examples of success and failure scenarios for the zero-gradient experiment where a blue arrow is a gradient direction in each step of PGD attack, and the red dash arrow is the distance between sample x to the zero-gradient location.



(a) Initial max = 0.01.



(b) Initial max = 0.1.



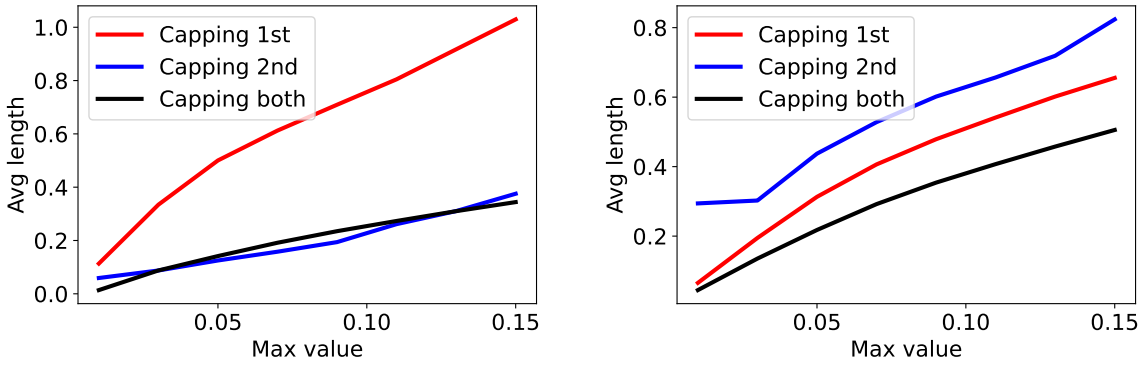
(c) Initial max = 1.

Figure 4.5. Average distance to zero gradients by PGD attack on a range of max values where the targets are general networks.

4.5 Experiments with Attacks

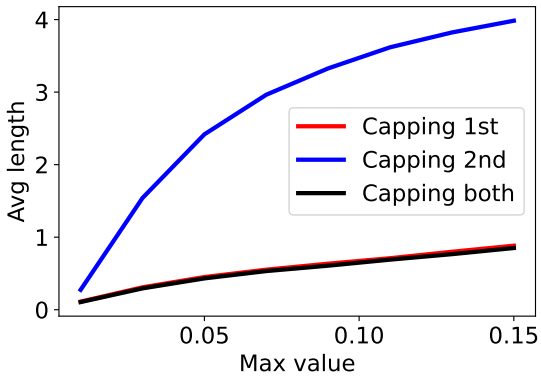
Next, in this section,[3] we apply some adversarial attacks to evaluate how much S-ReLU can be robust against them. Also, we compare the performance and robustness of S-ReLU with state-of-the-art approaches.

[3]Only S-ReLU with Adversarial Training subsection is in the published article; the rest is new unpublished material.



(a) Initial max = 0.01.

(b) Initial max = 0.1.



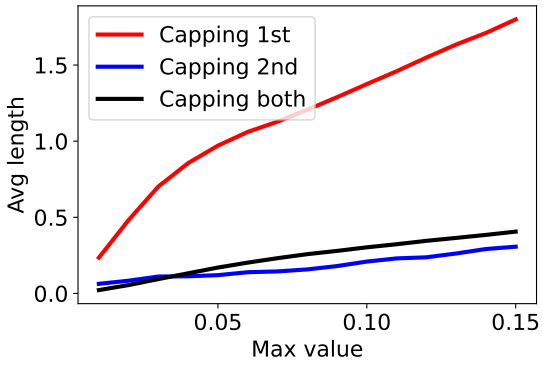
(c) Initial max = 1.

Figure 4.6. Average distance to zero-gradient areas by PGD attack on a range of max values where the targets are reversed networks with the MNIST dataset.

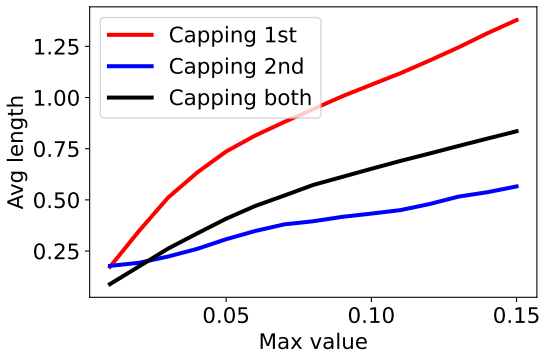
4.5.1 Datasets

We used MNIST (LeCun et al., 2010), FMNIST (Xiao et al., 2017b), KMNIST (Clanuwat et al., 2018) and EMNIST (Cohen et al., 2017) to show the empirical results for S-ReLU functions. We briefly describe these datasets here:

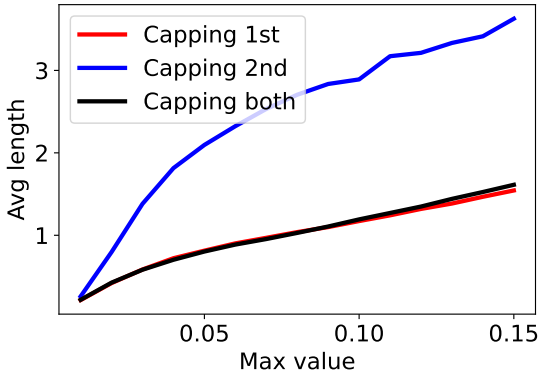
- **MNIST (Modified National Institute of Standards and Technology):** MNIST consists of 60000 training images and 10000 testing images, each 28x28 pixels, representing handwritten digits from 0 to 9. Figure 4.8 show some examples from this dataset.
- **Fashion-MNIST (FMNIST):** FMNIST has 60,000 training images and 10,000 testing images, each 28x28 pixels. It represents ten different fashion



(a) Initial max = 0.01.



(b) Initial max = 0.1.



(c) Initial max = 1.

Figure 4.7. Average distance to zero gradients by PGD attack on a range of max values where the targets are equal networks with the MNIST dataset.

categories, such as shoes, shirts, and dresses. FMNIST has 10 classes. Figure 4.9 show some examples from this dataset.

- **Kuzushiji-MNIST (KMNIST):** KMNIST comprises 60000 training images and 10000 testing images, each 28x28 pixels. It represents handwritten Japanese characters (Hiragana), and it has 10 classes. Figure 4.10 show some examples from this dataset.
- **Extended MNIST (EMNIST):** EMNIST is an extension of MNIST with additional variations. It has several types, and we use ByClass type which contains 814,255 characters. The number of classes is 62. EMNIST provides a diverse set of classes for character recognition challenges. Figure 4.11 show some examples from this dataset.

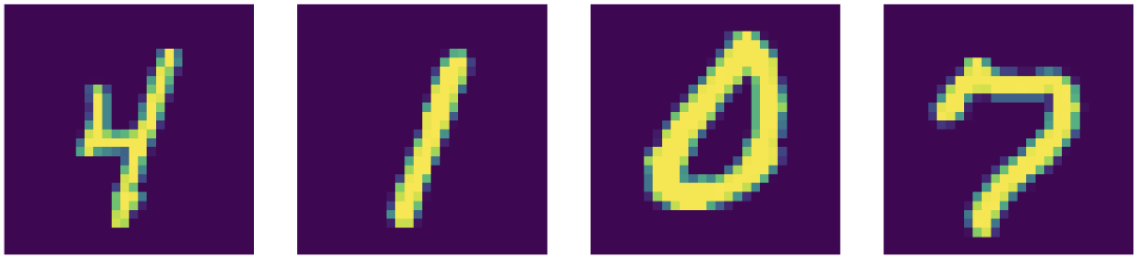


Figure 4.8. Examples of the MNIST dataset

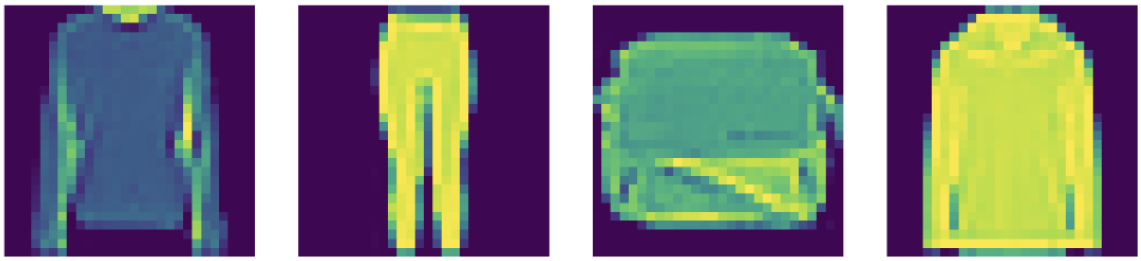


Figure 4.9. Examples of the FMNIST dataset

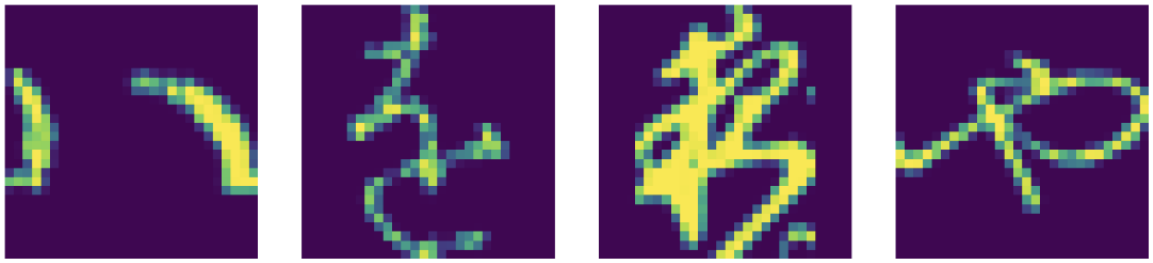


Figure 4.10. Examples of the KMNIST dataset

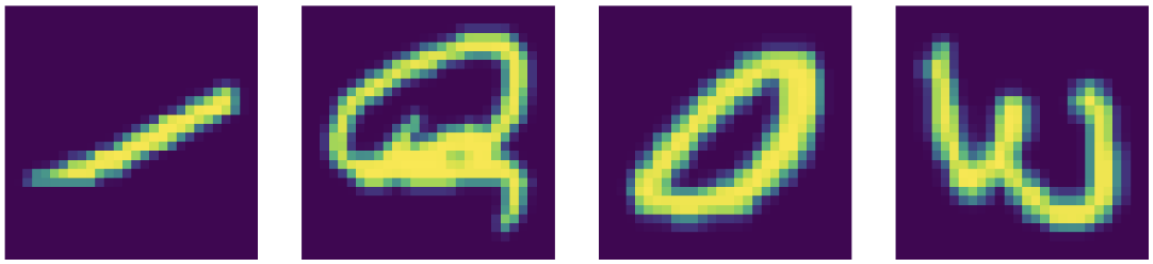


Figure 4.11. Examples of the EMNIST dataset

4.5.2 Training Details

We built two-hidden-layer models for this experiment and trained them with the Adam optimizer (Kingma and Ba, 2014). The first hidden layer has 392 neurons, and the second hidden layer has 196 neurons. At last, the output layer's size is the

number of classes in the dataset. We set the learning rate to 10^{-3} for 20 epochs. We ran the hyperparameter tuning through grid search and found that the model with these hyperparameters performed the best.

4.5.3 Adversarial Attacks

We utilized two adversarial attack strategies to generate adversarial examples from the test samples and compute the robust accuracy for trained targeted models. We used these two attacks because they are popular and widely used in the literature. These two attacks are:

- **Fast Gradient Sign Method (FGSM) (Goodfellow et al., 2014):** 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) (Madry et al., 2017):** 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.

4.5.4 Results

Table 4.2 shows our preliminary results for two-hidden-layer networks on MNIST, FMNIST, KMNIST, and EMNIST datasets. We only cap the ReLU function of the second hidden layer and demonstrate the results of different max values.

We follow ANOVA and obtain an F score of 11.15 while the critical value ($\alpha = 0.01$) is 4.05. Since the obtained F score is much greater than the critical value, we reject the null hypothesis. That is, there is a significant difference among the

classifiers. Also, when taking the average accuracy into consideration, capping ReLU functions can significantly improve the model's robustness.

We also use a non-parametric test, i.e., Friedman test, to measure the statistical difference between those results. We use average ranks in the last row of Table 4.2 in this test. As a result, we obtain the χ^2 score of 21.6, and the F_F score is 10.69 while the critical value ($\alpha = 0.01$) is around 4.126. Because those obtained scores are both greater than the critical value, we reject the null hypothesis. Therefore, the models are significantly different from each other with the confidence of 99%. After that, we use the Nemenyi test (Demšar, 2006) to identify which pair of classifiers is significantly different. We compute the critical difference and obtain 1.05. Hence, the classifiers with the max values of 0.1 and 0.01 are significantly more robust than the general classifier.

4.5.5 S-ReLU with Adversarial Training

In this section, we aim to investigate the efficacy of applying adversarial training techniques to S-ReLU classifiers to enhance their robustness beyond that of general classifiers that have undergone adversarial training. To accomplish this, we utilize two-hidden-layer neural networks as the base model and train them using clean test samples for a total of twenty epochs with the Adam optimizer as described in (Kingma and Ba, 2014) and a learning rate of 0.001. We only apply the ReLU function cap at the second hidden layer, as previous sections have demonstrated this to be the most effective location for such an operation.

Following the initial training phase, we then proceed to apply adversarial training to these networks through the use of either the Fast Gradient Sign Method (FGSM) or Projected Gradient Descent (PGD) for an additional ten epochs, with a perturbation bound of 0.1. Subsequently, we evaluate these networks' accuracy on clean test samples and samples that have been attacked using FGSM, PGD, and the Carlini and Wagner (CW) attack (Carlini and Wagner, 2017). For FGSM and PGD,

Table 4.2. Accuracy of MNIST, FMNIST, KMNIST and EMNIST two-hidden-layer classifiers with general ReLU and S-ReLU activation functions on clean test samples and adversarial test samples generated by using FGSM and PGD with two perturbation bounds (i.e., ϵ). Also, the average accuracy is provided. Note that the numbers in parentheses are ranks of the models based on their accuracy in each dataset and their averages are also provided.

Dataset	Attack	No Max %	Max = 1 %	Max = 0.1 %	Max = 0.01 %
MNIST	-	98.49 (1)	98.46 (2)	98.06 (3)	97.88 (4)
	FGSM ($\epsilon = 0.05$)	77.34 (4)	80.36 (3)	85.16 (2)	93.31 (1)
	PGD ($\epsilon = 0.05$)	55.92 (4)	56.47 (3)	67.90 (2)	93.13 (1)
	FGSM ($\epsilon = 0.1$)	41.77 (3)	41.24 (4)	68.04 (2)	92.37 (1)
	PGD ($\epsilon = 0.1$)	9.47 (3)	7.45 (4)	39.79 (2)	89.61 (1)
FMNIST	-	89.45 (2)	89.65 (1)	89.11 (3)	87.29 (4)
	FGSM ($\epsilon = 0.05$)	35.57 (4)	46.46 (3)	52.17 (2)	80.31 (1)
	PGD ($\epsilon = 0.05$)	18.62 (4)	22.16 (3)	40.96 (2)	83.84 (1)
	FGSM ($\epsilon = 0.1$)	17.64 (4)	26.52 (3)	38.23 (2)	79.06 (1)
	PGD ($\epsilon = 0.1$)	5.20 (4)	5.77 (3)	27.46 (2)	78.65 (1)
KMNIST	-	92.34 (1)	91.89 (2)	91.68 (3)	87.67 (4)
	FGSM ($\epsilon = 0.05$)	66.50 (4)	67.96 (3)	71.29 (2)	79.83 (1)
	PGD ($\epsilon = 0.05$)	50.44 (4)	50.93 (3)	56.15 (2)	82.08 (1)
	FGSM ($\epsilon = 0.1$)	30.20 (4)	34.65 (3)	48.89 (2)	76.96 (1)
	PGD ($\epsilon = 0.1$)	8.56 (4)	9.76 (3)	34.07 (2)	75.89 (1)
EMNIST	-	84.65 (2)	84.70 (1)	83.46 (3)	79.84 (4)
	FGSM ($\epsilon = 0.05$)	32.91 (4)	60.47 (3)	74.88 (2)	78.38 (1)
	PGD ($\epsilon = 0.05$)	8.40 (4)	40.14 (3)	79.13 (2)	79.23 (1)
	FGSM ($\epsilon = 0.1$)	10.60 (4)	31.79 (3)	59.00 (2)	76.54 (1)
	PGD ($\epsilon = 0.1$)	3.31 (4)	10.26 (3)	65.75 (2)	75.15 (1)
	Average	41.87 (3.4)	47.85 (2.8)	63.56 (2.2)	**83.35 (1.6)**

we employ a perturbation bound of 0.1, a maximum iteration of 10 and a step size of 0.01. Additionally, in the case of the CW attack, we use a maximum iteration of 10000, a learning rate of 0.01, an initial balancing factor of 0.001, and 9 adjustments of the balancing factor.

The configurations of the classifiers and their corresponding accuracy on both clean and adversarial test samples are presented in Table 4.3. The results reveal that by decreasing the maximum value, the robustness of the classifiers against attacks using FGSM, PGD, and CW can be improved without sacrificing a significant portion

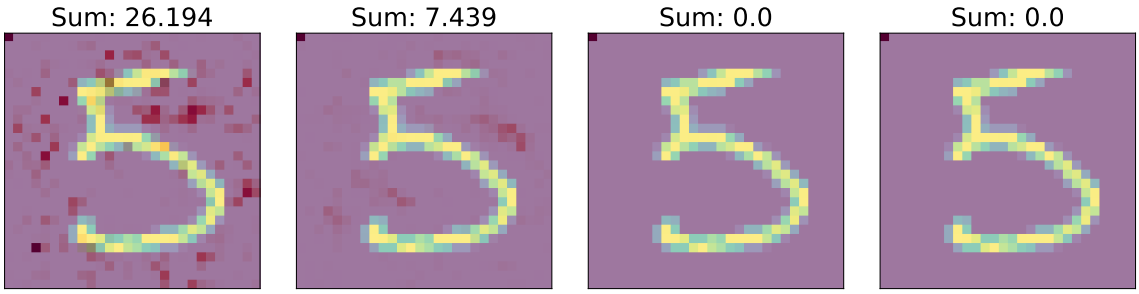
Table 4.3. Accuracy of MNIST two-hidden-layer classifiers with ReLU and S-ReLU on clean test samples and adversarial test samples generated by using FGSM, PGD and CW.

Max Val.	Adv. Training	Clean %	FGSM %	PGD %	CW(L_2) %
-	-	98.49	41.77	9.47	0.00
1.00	-	98.46	41.24	7.45	0.00
0.10	-	98.06	68.04	39.79	5.56
0.01	-	97.88	92.37	89.61	8.07
-	FGSM	98.26	91.44	85.12	0.19
1.00	FGSM	98.35	92.46	81.88	0.18
0.10	FGSM	98.18	93.00	90.37	3.50
0.01	FGSM	97.10	**94.07**	**96.36**	**8.21**
-	PGD	**98.67**	91.85	86.74	0.10
1.00	PGD	98.49	93.32	87.09	0.11
0.10	PGD	98.09	92.64	92.85	3.62
0.01	PGD	96.55	89.21	95.43	8.00

of standard accuracy. This is particularly evident when the models are retrained using FGSM, which results in similar performance and robustness to retraining using PGD, despite the latter taking much more time, as previously discussed in (Wong et al., 2020). Additionally, it is worth noting that although capping the ReLU function can improve robustness, the CW attack remains particularly effective, as it is not limited by any perturbation bound. Despite the success of the CW attack, we continue to see the trend that using a lower max value yields a more robust network. Therefore, a correctly customized classifier concerning its ReLU functions would ultimately be robust against CW. In this context, it is essential to note that static capping ReLU activation functions are a starting point for enhancing adversarial robustness by customizing architecture.

4.6 S-ReLU Classifier's Sensitivity Map

As discussed in (Sooksatra and Rivas, 2022), a pixel in an image vulnerable to adversarial attacks is sensitive to a slight change. In that work, we also proposed an



(a) General ReLU (b) Max val = 1 (c) Max val = 0.1 (d) Max val = 0.01

Figure 4.12. Sensitivity map of digit five and the summation of the scores on the top. Note that the more red pixel is, the more sensitive pixel becomes. Also, the black pixel in the top left of the image is not included in the map. We use it as a maximum reference value to tune the value's range across all the images.

equation to compute a sensitivity map to determine how much each pixel is susceptible to adversarial attacks. The equation is

$$\text{smap}(\boldsymbol{x}, Z) = \max\left(\mathbf{0}, \frac{\partial Z_t}{\partial \boldsymbol{x}} \cdot \sum_{c \neq t} \frac{\partial Z_c}{\partial \boldsymbol{x}}\right), \tag{4.3}$$

where $\boldsymbol{x}$ is an input, Z is a classifier whose output is before Softmax function, Z_i is the output of class i, t is the true class of $\boldsymbol{x}$ and $\mathbf{0}$ is a matrix of 0 whose size is the same as $\boldsymbol{x}$. We sum the map's values across all pixels to show that capping ReLU functions improves robustness.

We create a two-hidden-layer classifier and train it with several max values (i.e., 1, 0.1 and 0.01). Figure 4.12 shows the sensitivity map of digit five with the classifier. Essentially, the number of vulnerable pixels and the summation of the map decrease when the max value is reduced. Therefore, capping ReLU functions with low max values can improve the robustness.

4.7 Limitations

In this section, we discuss the limitations of S-ReLU.[4] While S-ReLU successfully enhances adversarial robustness in MNIST classifiers, their performance falters when applied to more extensive datasets like CIFAR-10. The challenge arises from the

[4]The material in this section does not appear in any of our other publications.

substantial layers and numerous zero gradients, leading to what is commonly known as the gradient vanishing problem. The upcoming sections will explain a compelling solution to address and overcome this issue, revolutionizing the capability of classifiers on larger datasets.

4.8 Conclusion

In this chapter, we provided a formal definition of the S-ReLU function and explored its optimal placement within a neural network to achieve the best balance between model performance and robustness. Our experiments indicate that integrating the S-ReLU function in smaller, deeper layers yields the most favorable trade-off. Additionally, the results from our zero-gradient experiments align closely with those from previous studies.

We also assess the impact of S-ReLU under adversarial conditions, employing attack methods such as the Fast Gradient Sign Method (FGSM) and Projected Gradient Descent (PGD). Our findings reveal that lowering the maximum value of S-ReLU enhances the model's defense against these attacks, albeit at a slight cost to performance. Furthermore, our statistical tests also indicate that S-ReLU significantly outperforms the general ReLU across several attacks. These observations are corroborated by the sensitivity analysis detailed in Sooksatra and Rivas (2022).

Despite the successes of S-ReLU in enhancing model robustness, one significant drawback is its limited generalizability to larger datasets. This limitation suggests an area for further research and potential improvement in the application of S-ReLU in diverse neural network architectures.

CHAPTER FIVE

Dynamic-Max-ReLU Activation Functions

In the preceding chapter, we demonstrated the effectiveness of S-ReLU in enhancing both model performance and adversarial robustness. We conducted a series of experiments that showcased improvements in resistance to adversarial attacks facilitated by the S-ReLU function. 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 chapter 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 in S-ReLU. However, the max values (i.e., m) of those functions are learnable. Therefore, at first, we set those values to be high and then try to minimize them during the training to improve the robustness such that the optimizer can adjust the models with the low max values. We minimize the max values because Table 4.1 shows that low max values (i.e., m) lead to small output differences and improve robustness. Therefore, the loss function can be formulated as

$$l(F(x,\theta),y) + \lambda \sum_i m_i^2, \tag{5.1}$$

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

5.1 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 on model robustness.

Firstly, we discuss the datasets employed in our experiments. These datasets have been 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 on 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.

5.1.1 Datasets

We used four datasets in this experiment: MNIST Deng (2012), CIFAR10 (Krizhevsky et al., 2009), CIFAR100 (Krizhevsky et al., 2009) and Tiny-Imagenet Le and Yang (2015). We already described MNIST in Chapter 4.5.1; CIFAR10 is a dataset commonly used for machine learning and computer vision tasks. CIFAR-10 consists of 60000 32x32 color images in 10 different classes, with each class representing a distinct object or animal category. The dataset is divided into 50000 training images and 10000 testing images. It is widely used as a benchmark for developing and evaluating image classification algorithms and models. Figure 5.1 shows some examples of this dataset.

The CIFAR100 dataset is a collection of 60000 32x32 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. The dataset is commonly used for evaluating machine learning algorithms and models due to its diverse set of classes and relatively small image size. Figure 5.2 shows some examples of this dataset.

TinyImageNet is a subset of the large-scale ImageNet dataset, designed for training deep neural networks on smaller computational resources. It consists of 200 diverse classes, with each class having 100000 training images and 10000 test images. Each image is of size 64x64 pixels and includes a wide range of object categories, making it a useful dataset for tasks like classification, detection, and segmentation. Tiny ImageNet 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 5.3 shows some examples of this dataset.

Figure 5.1. Examples of the CIFAR10 dataset

Figure 5.2. Examples of the CIFAR100 dataset

Figure 5.3. Examples of the Tinyimagenet dataset

5.1.2 Training Details

The optimization process employed the Adam optimizer (Kingma and Ba, 2014) with an initial learning rate set to 10^{-3}. Additionally, we implemented the *ReduceOnPlateau* 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 *ReduceOnPlateau* callback reduces the learning by multiplying it with 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 *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 add a dense layer before the output layer. Incorporating a dense layer before the output layer is motivated by findings from the experiment conducted in (Sooksatra et al., 2023). The study demonstrates 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 5.4 while it is a general ReLU for other approaches.

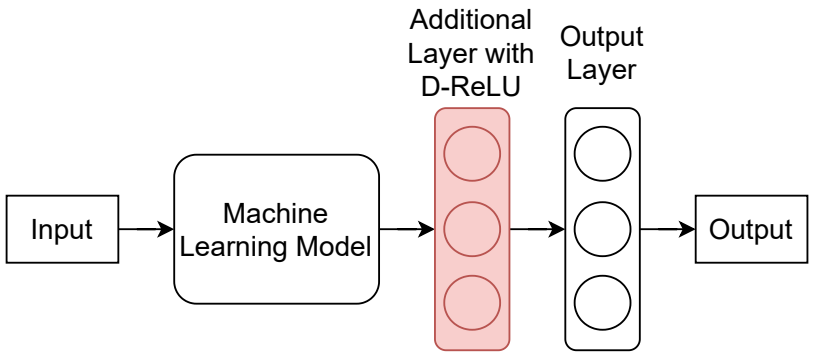


Figure 5.4. Architecture of our approach by adding a layer (in red) with D-ReLU before the output layer

5.1.3 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) (Goodfellow et al., 2014) and Projected Gradient Descent (PGD) (Madry et al., 2017):** As elucidated in Chapter 4.5.3, these attacks form integral components of our evaluation framework.
- **Auto Projected Gradient Descent with Cross Entropy Loss (APGD_CE) (Croce and Hein, 2020):** Similar to PGD, this attack

leverages the cross entropy loss function to generate adversarial examples. Notably, it incorporates an adaptive step size, distinguishing it from PGD.

- **Auto Projected Gradient Descent with DLR (APGD_DLR) (Croce and Hein, 2020):** This variant of APGD_CE maintains the same underlying principles as APGD_CE but employs Difference of Logits Ratio Loss (DLR) (Croce and Hein, 2020) as the loss function.
- **Carlini and Wagner Attack with L2 Norm (CW_L2) (Carlini and Wagner, 2017):** 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 L2 norm and maximizes the misclassification confidence as well.
- **Square (Andriushchenko et al., 2020):** This attack is blackbox and 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 from 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.

5.1.4 SOTA Methods for Robustness

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

- **Adversarial Training (Madry et al., 2017):** This method retrains a model with adversarial examples after its successful natural training. We retrained the models for 10 epochs.
- **TRADES (Zhang et al., 2019):** 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 the method utilizes a parameter denoted as β to strike a balance between performance and robustness. We adopted the same values of β as employed by the original authors, specifically $\beta = 1$ and 6.

We used PGD for generating adversarial examples for all the mentioned methods.

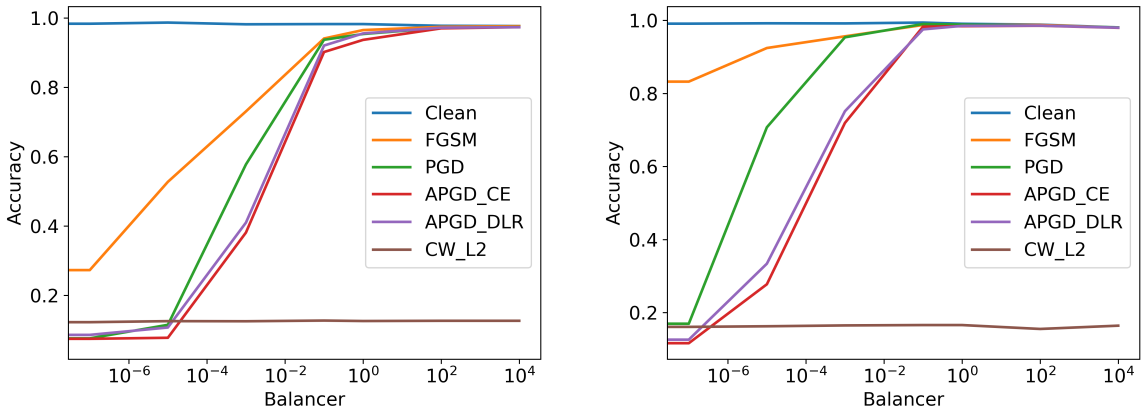
5.2 Whitebox-Attack Experiments

5.2.1 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 for FGSM, PGD, PGD_CE and PGD_DLR to be 0.1. Also, we set the perturbation bound for CW_L2 to be 18.

The outcomes of tuning the balancer, denoted as λ in (5.1), are illustrated in Figure 5.5. Note that at the balancer of zero, the models were naturally trained, and they were not robust against the 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 perturbation applied may remain consistent across all samples.



(a) Dense

(b) Shallow CNN

Figure 5.5. Accuracy of two types of networks on clean MNIST and adversarial examples when adding the dense layer with a D-ReLU function before the output layer.

Table 5.1 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 select the optimal tradeoff 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 5.1. Accuracy metrics for dense networks and shallow CNNs under various robust training schemes, evaluating on both clean samples and adversarial examples generated by different attacks on the MNIST dataset. Note that the accuracy metrics in bold are the highest in a specific model among the different training methods. 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 parenthesis are the ranks for training methods under an architecture, TRADES-k means the TRADES approach with $\beta = k$, and D-ReLU-k means the D-ReLU approach with $m = k$.

Model	Training	Clean %	FGSM %	PGD %	AP_{CE} %	AP_{DLR} %	CW_{L2} %
Dense	AT	**98.10 (1)**	89.77 (4)	87.7 (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)**

5.2.2 Experimental Results for CIFAR10

We trained 6 types of models: two-hidden-layer dense networks, shallow convolutional neural networks (CNN), ResNet50 (He et al., 2016), ResNet101 (He et al., 2016), MobilenetV2 (Sandler et al., 2018) and InceptionV3 (Szegedy et al., 2016). We set the perturbation bounds for FGSM, PGD, APGD_CE and APGE_DLR to be 0.01. Moreover, the bound for CW_L2 is 18.

Figure 5.6 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

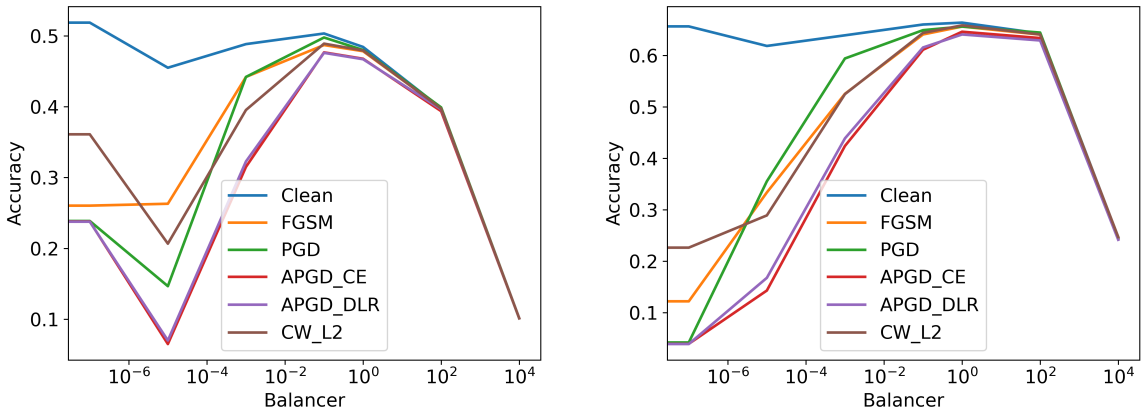
these models withstand adversarial perturbations when adjusted with varying levels of balancers.

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 diminished 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 for 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.

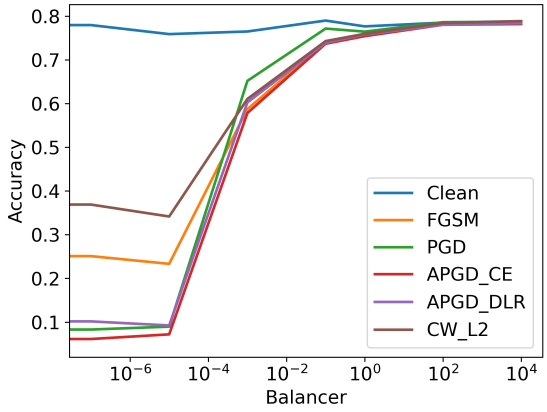
The implications of these results are multifaceted. Firstly, they underscore the importance of balancing robustness and performance. While enhancing defense mechanisms against adversarial attacks is crucial, maintaining high levels of accuracy and performance in non-adversarial scenarios is equally important. 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

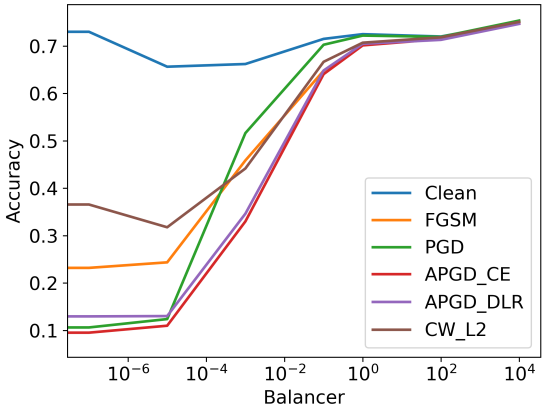


(a) Dense

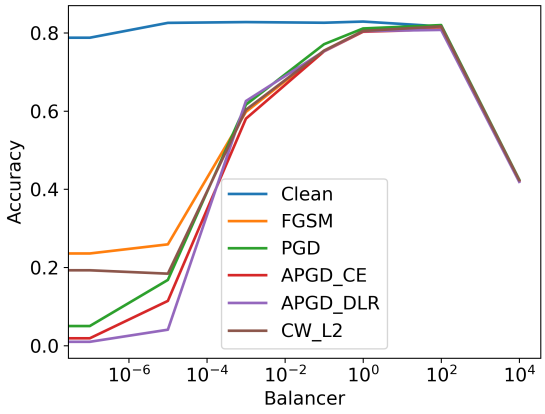
(b) Shallow CNN



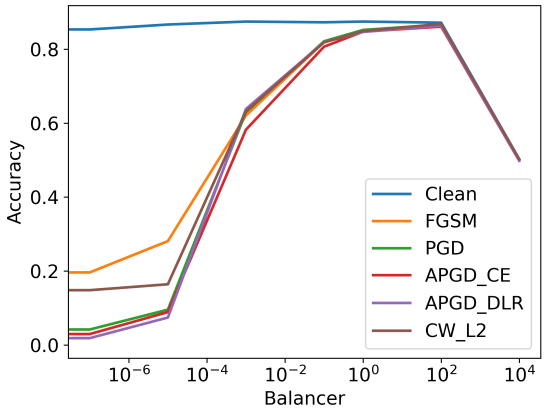
(c) ResNet50



(d) ResNet101



(e) MobilenetV2



(f) InceptionV3

Figure 5.6. Accuracy of several types of networks on clean CIFAR10 and adversarial examples when adding the dense layer with a D-ReLU function before the output layer.

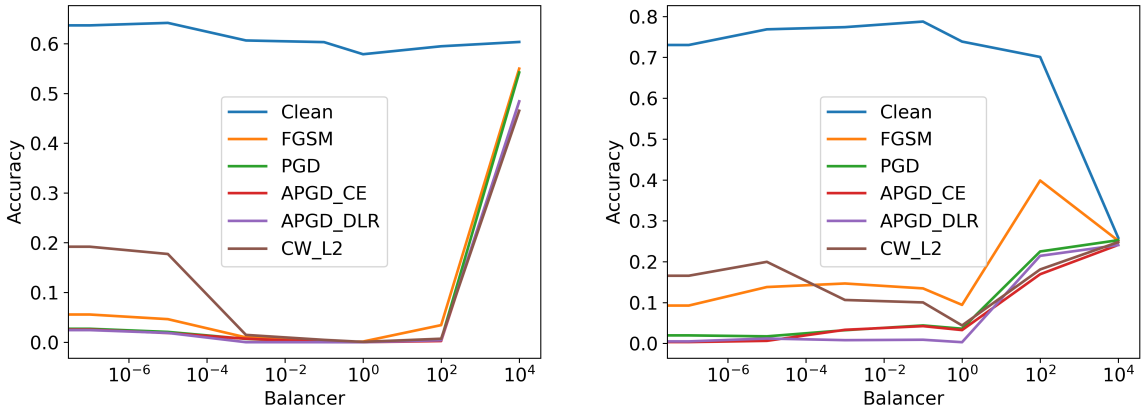
model's general performance. Identifying this optimal range can guide the development of more resilient machine learning systems.

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 5.7 presents the outcomes, illustrating the impact on several CNN architectures when the D-ReLU layer is added at the beginning of the network.

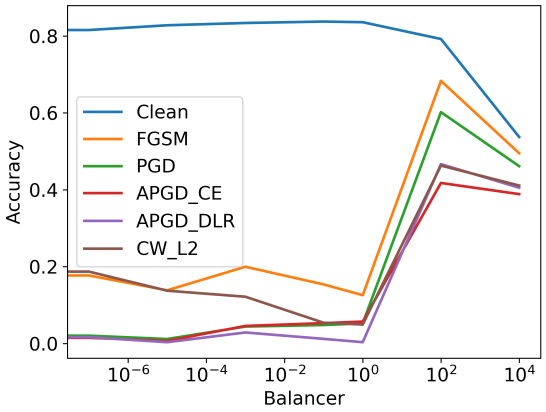
The results indicate that positioning the D-ReLU layer early in the network does not yield the same level of effectiveness as when placed in deeper layers. For the Shallow CNN (Figure 5.7a), MobilenetV2 (Figure 5.7b), and InceptionV3 (Figure 5.7c), there is a notable decline in adversarial robustness across different attack types (FGSM, PGD, APGD_CE, APGD_DLR, 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 designing robust neural network architectures. Incorporating D-ReLU in deeper layers allows the network to better leverage its properties for improving 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.



(a) Shallow CNN

(b) MobilenetV2



(c) InceptionV3

Figure 5.7. 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.

Table 5.2 and A.1 provide 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.

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 for computing adversarial examples

Table 5.2. Accuracy metrics for multiple types of networks under various robust training schemes, evaluating 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, TRADES-k means the TRADES approach with $\beta = k$, and D-ReLU-k means the D-ReLU approach with $m = k$.

Model	Training	Clean %	FGSM %	PGD %	AP_{CE} %	AP_{DLR} %	CW_{L2} %
Dense	AT	**52.33**	34.20	32.83	32.73	31.80	40.10
	TRADES-1	52.32	29.97	29.23	29.20	28.37	38.67
	TRADES-6	51.30	**37.00**	**36.53**	**36.50**	**34.57**	**42.30**
	D-ReLU-10^{-7}	51.87	26.03	23.87	23.80	23.77	36.10
Shallow CNN	AT	67.13	42.83	40.07	39.90	38.37	50.67
	TRADES-1	**67.37**	38.83	35.93	35.97	34.13	48.60
	TRADES-6	63.47	46.13	44.80	44.80	42.67	51.67
	D-ReLU-10^{0}	66.37	**65.60**	**65.60**	**64.60**	**64.07**	**65.83**
Resnet 50	AT	78.20	54.77	49.37	48.90	49.97	63.00
	TRADES-1	75.63	52.12	40.77	39.87	40.20	56.43
	TRADES-6	71.63	54.20	50.90	50.40	48.23	57.63
	D-ReLU-10^{4}	**78.87**	**78.83**	**78.73**	**78.20**	**78.40**	**78.87**
Resnet 101	AT	68.90	44.90	40.33	39.43	38.27	49.30
	TRADES-1	74.60	47.07	32.87	31.17	31.37	51.40
	TRADES-6	66.67	45.43	39.80	39.17	35.93	47.67
	D-ReLU-10^{4}	**75.10**	**75.03**	**75.37**	**74.73**	**74.67**	**75.10**
Mobilenet V2	AT	77.97	46.50	32.93	30.73	32.10	51.80
	TRADES-1	73.13	46.23	31.00	28.87	28.77	49.37
	TRADES-6	68.40	48.80	43.23	43.03	40.80	51.13
	D-ReLU-10^{2}	**81.67**	**81.57**	**82.00**	**80.87**	**80.77**	**81.67**
Inception V3	AT	84.60	64.27	58.80	58.30	59.33	66.47
	TRADES-1	82.53	62.30	52.67	51.90	51.87	62.40
	TRADES-6	76.97	61.97	58.00	57.80	56.03	62.10
	D-ReLU-10^{2}	**87.17**	**86.70**	**86.57**	**86.13**	**86.23**	**86.83**

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 preserved 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 perform an ANOVA test and obtain an F score of 17.4, which surpasses the critical value of 3.92 at $\alpha = 0.01$. Given that the F score is 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 the model's robustness.

Furthermore, We conduct a non-parametric test, specifically the Friedman test, to assess the differences between the results. This test uses the ranks provided in Table A.1. The test yields a χ^2 score of 39.43 and an F_F score of 20.12. The critical value ($\alpha = 0.01$) ranges between 2.13 and 2.18 for the degrees of freedom of 3 and 105, respectively. Given that both the chi-square and F_F scores are 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 employ the Nemenyi test (Demšar, 2006) to pinpoint which pairs of classifiers exhibit significant differences. The computed critical difference is 0.656. The differences in the average ranks between D-ReLU and the other techniques are 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.

5.2.3 Experimental Results for CIFAR100

Figure 5.8 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.

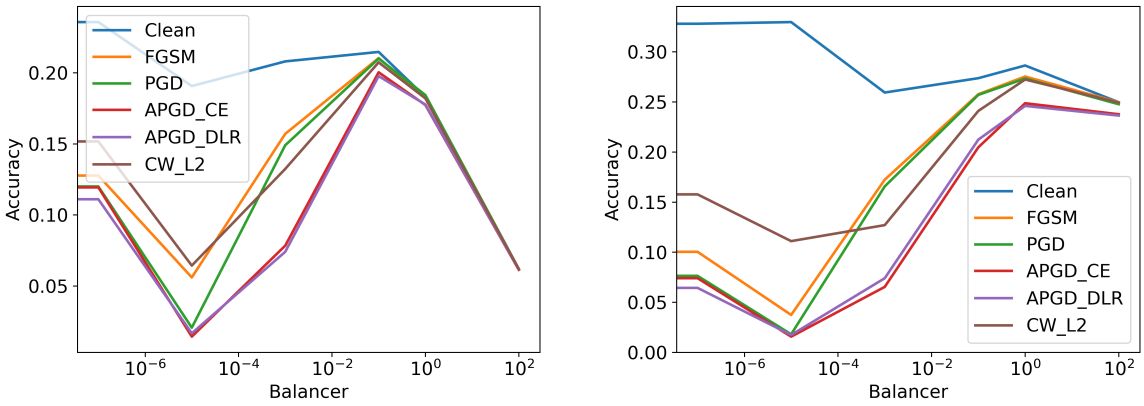
There is a notable improvement in robustness with increasing balancer values for adversarial examples. This trend is consistent across all types of white-box attacks considered: FGSM, PGD, APGD_CE, APGD_DLR, and CW_L2. The accuracy on adversarial examples shows a significant upward trajectory, especially 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

Table 5.3 and A.2 show the 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.

5.2.4 Experimental Results for TinyImagenet

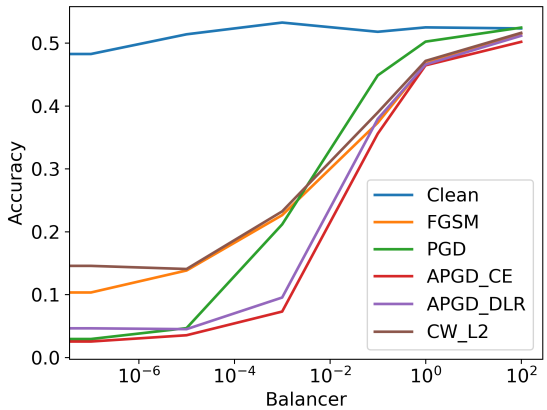
Figure 5.9 presents the accuracy of several neural network architectures on clean TinyImagenet samples and adversarial examples produced by various white-box attacks. The networks assessed include Dense, Shallow CNN, ResNet50, ResNet101, MobilenetV2, and InceptionV3. The experiments involve integrating 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

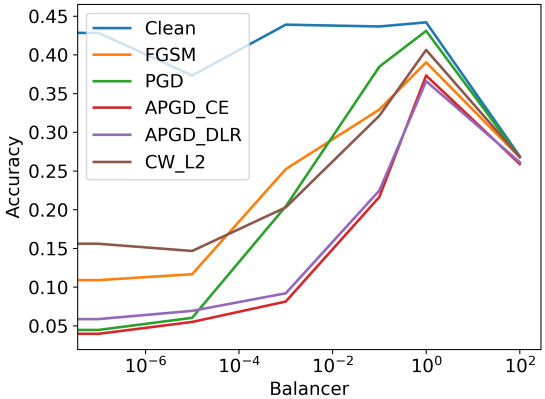


(a) Dense

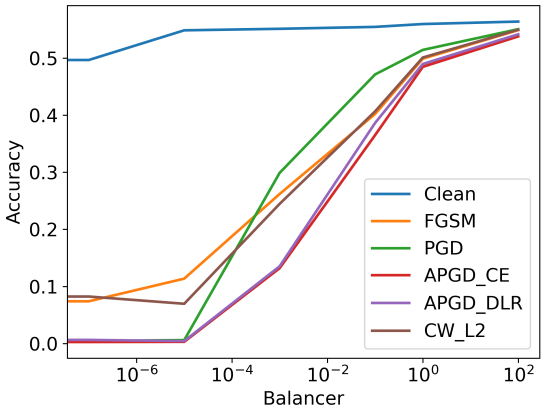
(b) Shallow CNN



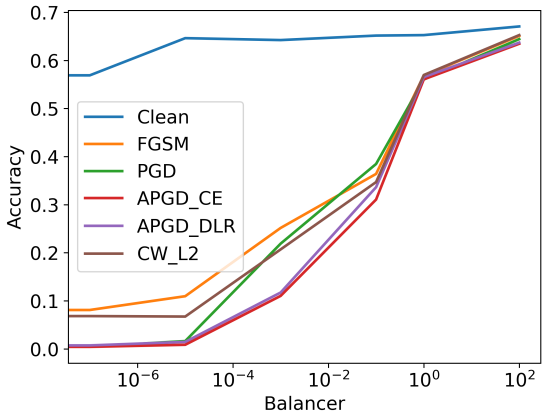
(c) ResNet50



(d) ResNet101

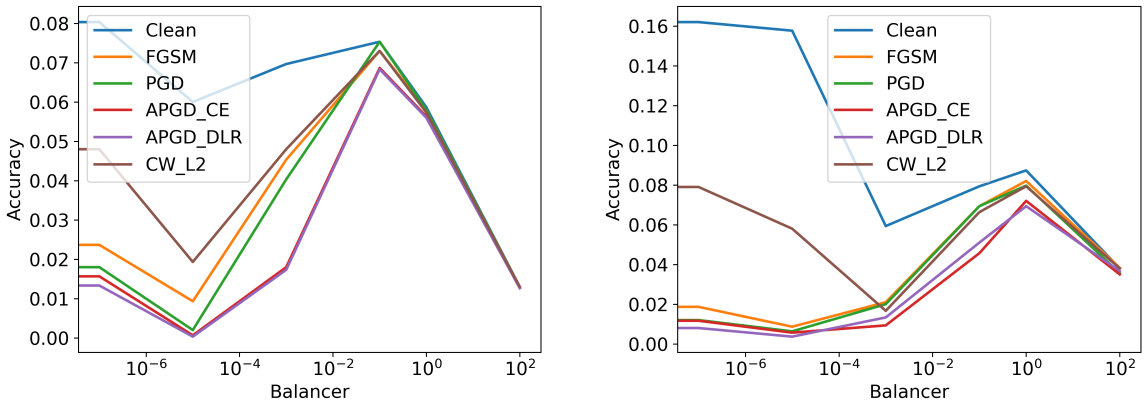


(e) MobilenetV2



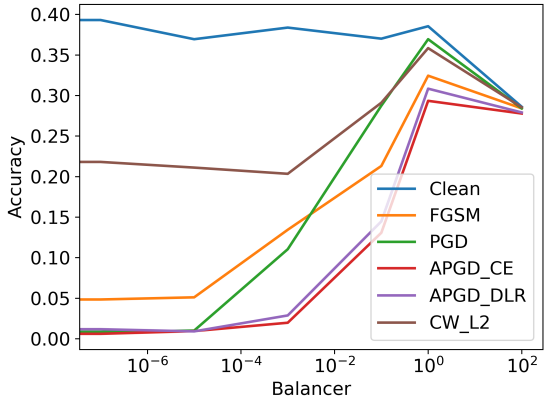
(f) InceptionV3

Figure 5.8. Accuracy of several types of networks on clean CIFAR100 and adversarial examples when adding the dense layer with a D-ReLU function before the output layer.

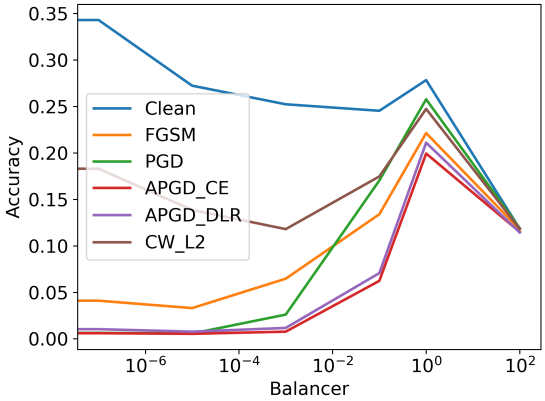


(a) Dense

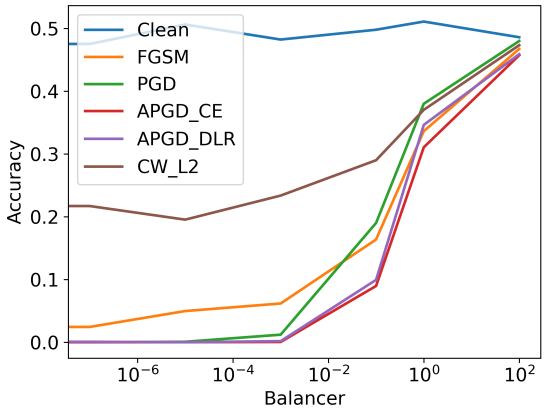
(b) Shallow CNN



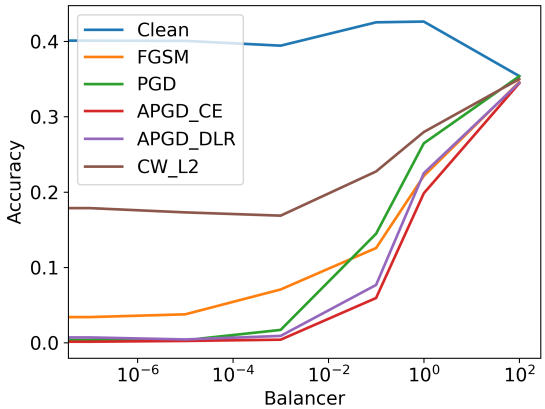
(c) ResNet50



(d) ResNet101



(e) MobilenetV2



(f) InceptionV3

Figure 5.9. Accuracy of several types of networks on clean TinyImagenet and adversarial examples when adding the dense layer with a D-ReLU function before the output layer.

Table 5.3. Accuracy metrics for multiple types of networks under various robust training schemes, evaluating 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, TRADES-k means the TRADES approach with $\beta = k$, and D-ReLU-k means the D-ReLU approach with $m = k$.

Model	Training	Clean %	FGSM %	PGD %	AP_{CE} %	AP_{DLR} %	CW_{L2} %
Dense	AT	**24.47**	14.80	14.30	14.20	12.63	17.53
	TRADES-1	22.97	13.37	13.23	13.17	11.30	16.27
	TRADES-6	23.27	13.87	13.73	13.60	12.27	16.60
	D-ReLU-10^{-1}	21.47	**21.03**	**21.00**	**20.03**	**19.77**	**20.73**
Shallow CNN	AT	**37.03**	17.73	16.47	16.30	14.43	22.30
	TRADES-1	32.60	12.87	11.50	11.43	9.47	18.50
	TRADES-6	34.80	18.67	17.87	17.87	15.40	22.33
	D-ReLU-1	28.63	**27.53**	**27.33**	**24.87**	**24.60**	**27.23**
Resnet 50	AT	48.67	26.67	21.83	21.53	23.13	31.90
	TRADES-1	48.97	26.57	19.80	19.27	20.03	30.50
	TRADES-6	43.97	28.90	26.03	25.70	24.03	30.63
	D-ReLU-10^{2}	**52.33**	**51.53**	**52.47**	**50.20**	**51.17**	**51.63**
Resnet 101	AT	44.97	23.57	18.67	18.33	18.77	27.77
	TRADES-1	**48.10**	24.17	17.70	16.87	17.80	28.10
	TRADES-6	45.20	28.21	20.53	19.32	19.44	30.02
	D-ReLU-1	44.20	**39.03**	**43.10**	**37.33**	**36.60**	**40.63**
Mobilenet V2	AT	51.37	23.83	15.30	14.43	15.73	28.50
	TRADES-1	42.97	19.50	9.47	8.20	8.60	20.70
	TRADES-6	40.13	24.50	20.73	20.13	18.87	25.40
	D-ReLU-1	**56.40**	**54.90**	**55.07**	**53.80**	**54.17**	**54.97**
Inception V3	AT	56.37	32.57	27.20	26.60	28.80	34.33
	TRADES-1	60.63	35.63	26.80	25.83	26.50	35.07
	TRADES-6	51.10	34.43	31.20	30.90	29.50	34.47
	D-ReLU-10^{2}	**67.07**	**65.10**	**64.43**	**63.47**	**63.70**	**65.27**

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.

Table 5.4 shows the performance and robustness of our approach and the other baselines on the TinyImagenet dataset. Also, Table A.3 shows the ranking of the approaches in each architecture. Our approach is struggling to find the balance between performance and robustness. However, in MobilenetV2, our approach outperforms the other ones in terms of performance and robustness.

5.2.5 Discussion

The consistent improvements in adversarial robustness across 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 the 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

Table 5.4. Accuracy metrics for multiple types of networks under various robust training schemes, evaluating 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, TRADES-k means the TRADES approach with $\beta = k$, and D-ReLU-k means the D-ReLU approach with $m = k$.

Model	Training	Clean %	FGSM %	PGD %	AP_{CE} %	AP_{DLR} %	CW_{L2} %
Dense	AT	8.63	5.40	5.13	5.00	4.27	7.00
	TRADES-1	8.57	4.80	4.77	4.73	4.10	**7.47**
	TRADES-6	**8.70**	5.07	5.13	5.10	3.93	7.30
	D-ReLU-10^{-1}	7.53	**7.30**	**7.53**	**6.87**	**6.83**	7.30
Shallow CNN	AT	**18.33**	4.80	4.17	4.10	2.73	10.60
	TRADES-1	14.93	2.17	1.60	1.60	0.97	8.10
	TRADES-6	16.37	4.57	4.07	3.97	2.67	**10.63**
	D-ReLU-1	8.40	**8.20**	**7.97**	**7.20**	**6.93**	7.93
Resnet 50	AT	40.67	17.57	13.17	12.93	14.03	30.87
	TRADES-1	**48.10**	22.15	16.10	15.55	14.95	**36.35**
	TRADES-6	40.97	23.93	21.87	21.57	19.77	31.57
	D-ReLU-1	38.53	**32.43**	**36.93**	**29.33**	**30.83**	35.83
Resnet 101	AT	32.73	15.43	13.10	12.63	11.40	24.17
	TRADES-1	**47.57**	20.50	15.07	14.57	14.43	**34.73**
	TRADES-6	39.13	**22.30**	20.37	**20.03**	17.67	30.63
	D-ReLU-1	27.83	22.13	**25.77**	19.93	**21.10**	24.73
Mobilenet V2	AT	50.00	23.13	16.73	16.30	16.97	**37.73**
	TRADES-1	48.87	20.60	13.57	12.83	12.00	35.10
	TRADES-6	43.20	23.70	21.23	20.87	19.03	33.73
	D-ReLU-1	**51.10**	**33.63**	**38.00**	**31.07**	**34.63**	37.03
Inception V3	AT	39.07	18.67	14.63	14.57	15.20	27.90
	TRADES-1	**60.43**	**32.53**	23.37	22.67	24.13	**46.17**
	TRADES-6	50.43	32.03	**29.23**	**28.90**	**28.30**	40.40
	D-ReLU-1	42.63	22.13	26.47	19.83	22.50	27.97

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 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, thus enhancing their reliability and security in various applications.

5.3 Blackbox-Attack Experiments

In addition to the promising results against white-box attacks, we have also evaluated the performance of the D-ReLU function in enhancing the robustness of CNNs against black-box attacks, specifically the Square attack. Figure 5.10, 5.11 and 5.12 provided offer valuable insights into how D-ReLU impacts various models across different datasets under black-box attack scenarios.

5.3.1 Experimental Results for CIFAR10

In Figure 5.10, the accuracy of several network types on clean CIFAR10 data and adversarial examples generated by the blackbox attack is depicted. For dense networks (Figure 5.10a), 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 5.10b) display a similar pattern, with a significant improvement in adversarial robustness at higher balancer values, while the clean accuracy remains consistent.

ResNet50 and ResNet101 (Figures 5.10c and 5.10d) 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 5.10e) also shows consistent improvement in adversarial accuracy with higher balancer values, despite slight fluctuations in clean accuracy. InceptionV3 (Figure 5.10f) exhibits a strong increase in adversarial robustness with higher balancer values while maintaining high accuracy on clean samples.

5.3.2 Experimental Results for CIFAR100

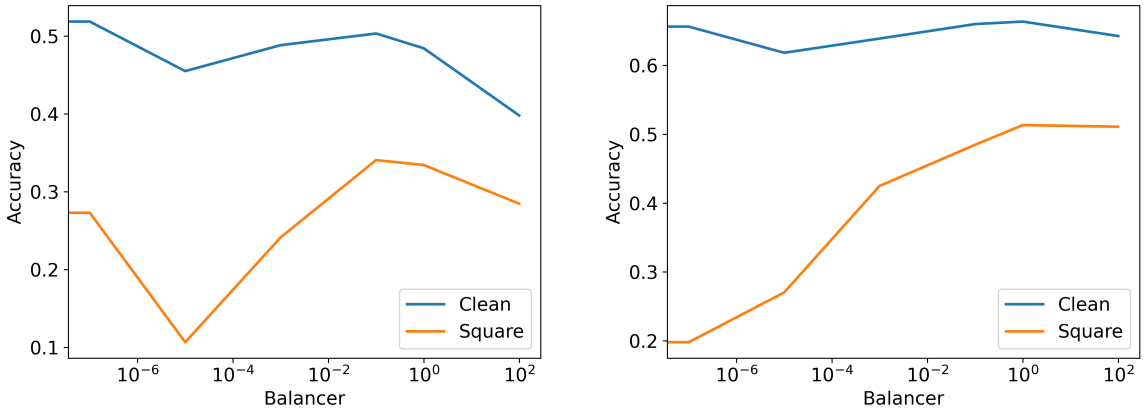
Figure 5.11 presents the accuracy metrics for CIFAR100. Dense networks (Figure 5.11a) show moderate improvement in adversarial robustness with the addition of the D-ReLU layer, though clean accuracy remains largely unaffected. Shallow CNNs (Figure 5.11b) follow a clear trend of increasing adversarial accuracy with higher balancer values, indicating the D-ReLU layer's effectiveness in enhancing robustness.

For deeper networks like ResNet50 and ResNet101 (Figures 5.10c and 5.11d), there is improved adversarial robustness with increasing balancer values, though a slight decrease in clean accuracy is observed at higher balancer values. MobilenetV2 (Figure 5.11e) displays marked improvement in adversarial robustness with higher balancer values, with minimal fluctuations in clean accuracy. InceptionV3 (Figure 5.11f) shows the highest gains in adversarial robustness, maintaining strong performance on clean samples.

5.3.3 Experimental Results for TinyImagenet

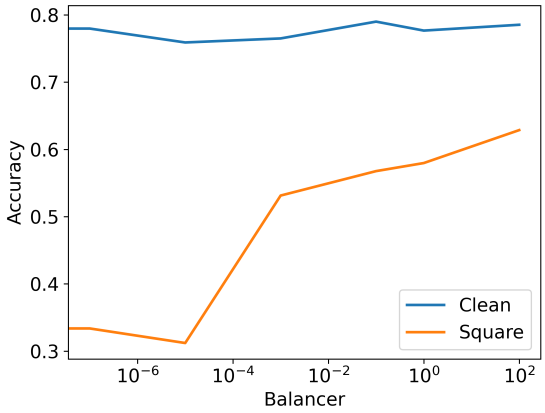
In Figure 5.12, the results for TinyImagenet are detailed. Dense networks (Figure 5.12a) show a significant increase in adversarial robustness with higher balancer values, while clean accuracy remains stable. Shallow CNNs (Figure 5.12b) exhibit improved adversarial accuracy with higher balancer values, though clean accuracy shows some variability.

Deeper networks like ResNet50 and ResNet101 (Figures 5.12c and 5.12d) benefit significantly in terms of adversarial robustness with increasing balancer values, with

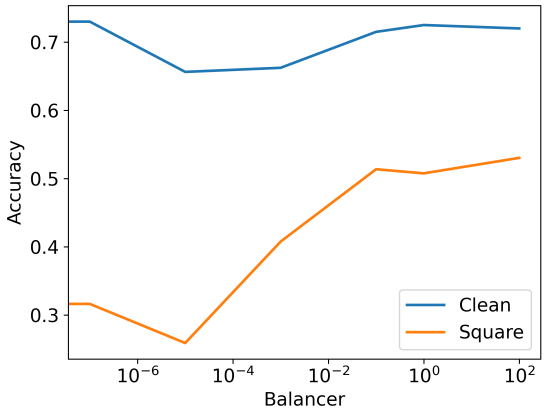


(a) Dense

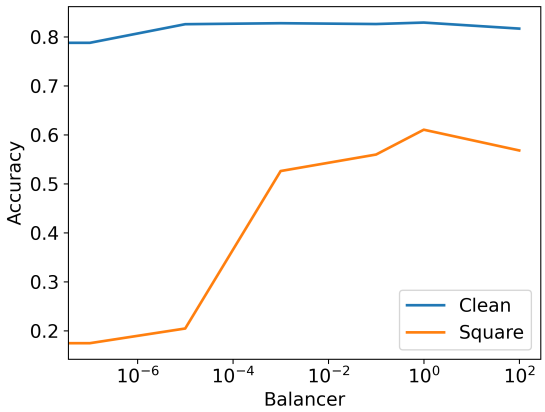
(b) Shallow CNN



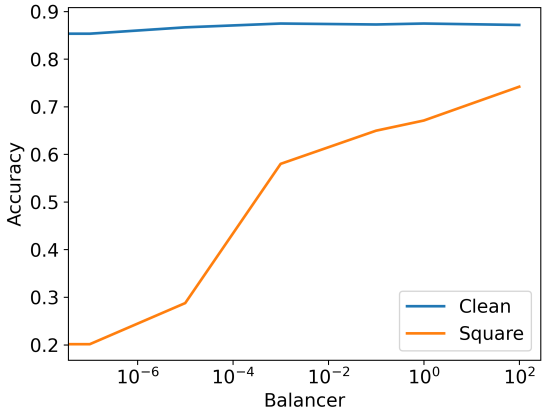
(c) ResNet50



(d) ResNet101

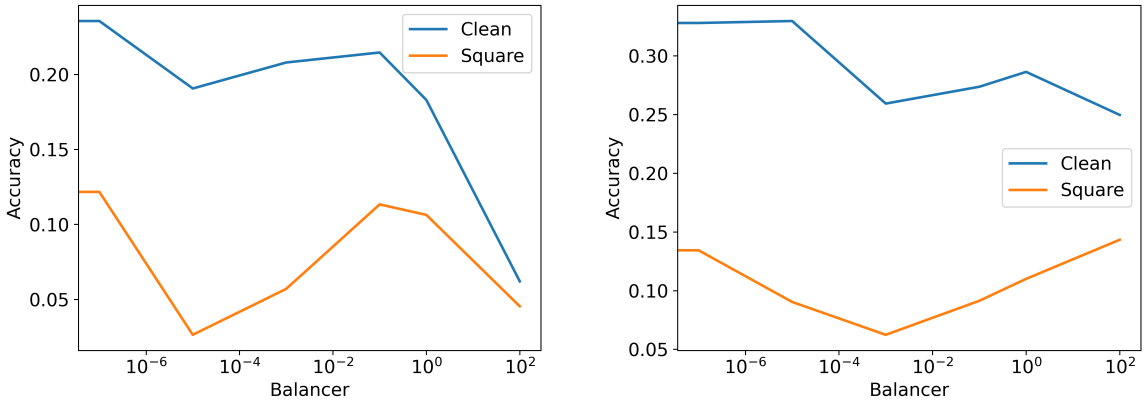


(e) MobilenetV2



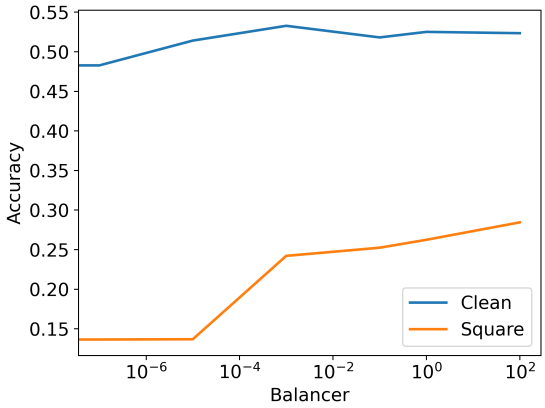
(f) InceptionV3

Figure 5.10. Accuracy of several types of networks on clean CIFAR10 and adversarial examples generated by a blackbox attack (i.e., square attack) when adding the dense layer with a D-ReLU function before the output layer.

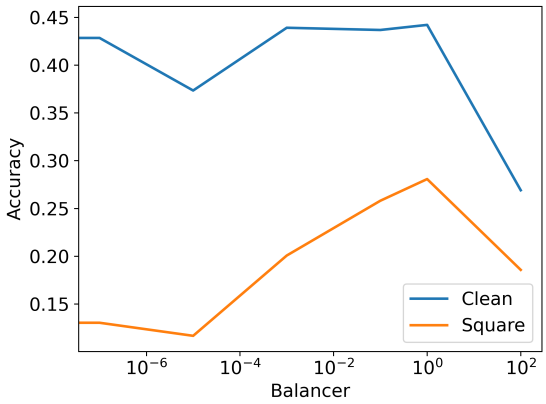


(a) Dense

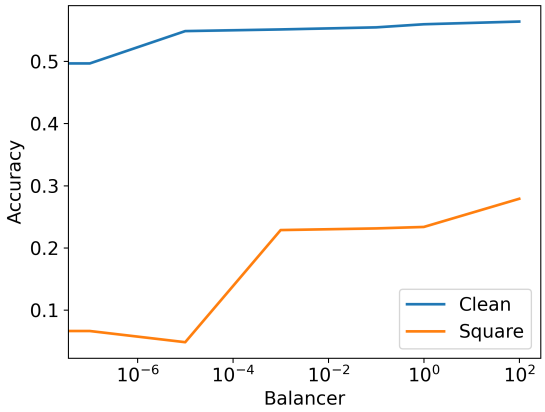
(b) Shallow CNN



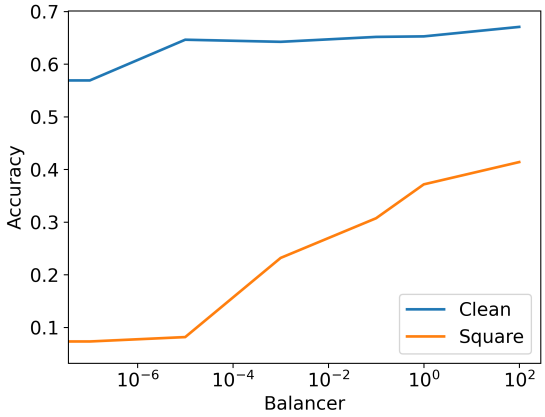
(c) ResNet50



(d) ResNet101



(e) MobilenetV2



(f) InceptionV3

Figure 5.11. Accuracy of several types of networks on clean CIFAR100 and adversarial examples generated by a blackbox attack (i.e., square attack) when adding the dense layer with a D-ReLU function before the output layer.

slight fluctuations in clean accuracy. MobilenetV2 (Figure 5.12e) demonstrates notable improvement in adversarial robustness with higher balancer values, with clean accuracy remaining relatively unaffected. InceptionV3 (Figure 5.12f) shows the most substantial gains in adversarial robustness among all tested architectures, with clean accuracy remaining high.

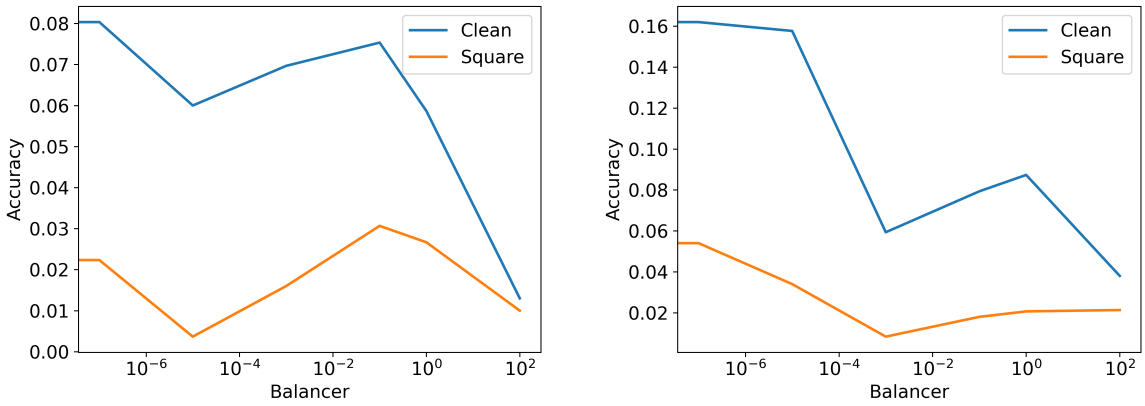
5.3.4 Comparison to Other Baselines

Table 5.5 and A.4 provide 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 blackbox attack (denoted as Square) on the CIFAR10, CIFAR100, and TinyImagenet datasets. The values displayed are in 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 Resnet50, MobilenetV2, and InceptionV3 models, D-ReLU stands out as the top performer on 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.

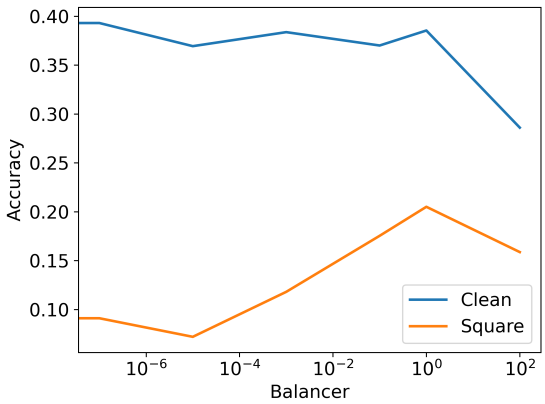
5.3.5 Discussion

The effectiveness of D-ReLU against the black-box attack 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

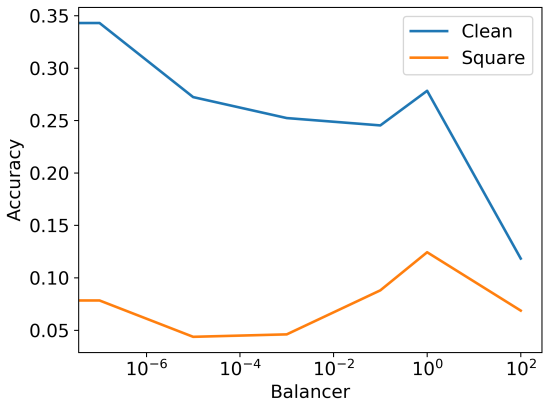


(a) Dense

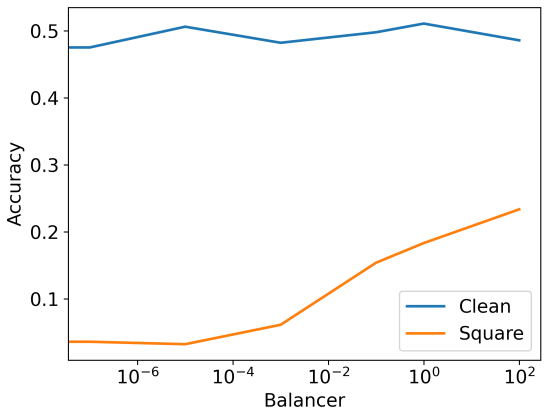
(b) Shallow CNN



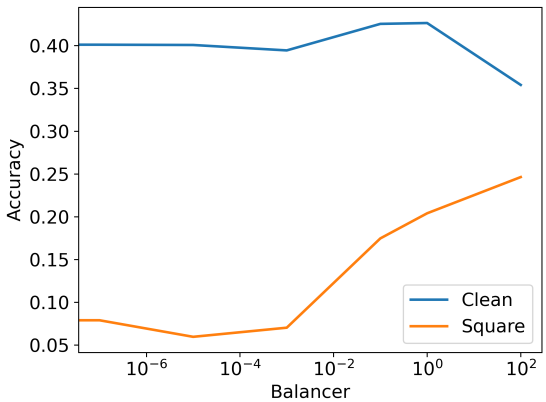
(c) ResNet50



(d) ResNet101



(e) MobilenetV2



(f) InceptionV3

Figure 5.12. Accuracy of several types of networks on clean Tinyimagenet and adversarial examples generated by a blackbox attack (i.e., square attack) when adding the dense layer with a D-ReLU function before the output layer.

Table 5.5. Accuracy metrics for multiple types of networks under various robust training schemes, evaluating on both clean samples and adversarial examples generated by a blackbox 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. Note that TRADES-k means the TRADES approach with $\beta = k$.

Model	Training	CIFAR10		CIFAR100		TinyImagenet	
		Clean %	Square %	Clean %	Square %	Clean %	Square %
Dense	TRADES-1	**52.33**	34.03	22.97	13.90	8.57	4.80
	TRADES-6	51.30	**38.47**	**23.27**	**14.13**	**8.70**	**4.87**
	D-ReLU	48.43	33.43	21.47	11.33	7.53	3.07
Shallow CNN	TRADES-1	67.37	45.93	32.60	15.47	14.93	5.43
	TRADES-6	64.50	49.30	**34.80**	**19.70**	**16.37**	**7.13**
	D-ReLU	**66.37**	**51.33**	32.87	13.53	16.20	5.40
Resnet50	TRADES-1	75.70	50.70	48.97	25.10	**48.40**	**26.53**
	TRADES-6	71.63	53.57	43.97	27.03	40.97	25.03
	D-ReLU	**78.53**	**62.87**	**52.33**	**28.43**	38.53	20.50
Resnet101	TRADES-1	**74.60**	45.37	**48.10**	23.20	**47.57**	**25.07**
	TRADES-6	66.67	43.63	10.67	1.67	39.13	24.00
	D-ReLU	72.00	**53.03**	44.20	**28.07**	27.83	12.43
MobilenetV2	TRADES-1	73.13	43.13	42.97	15.40	48.87	25.00
	TRADES-6	68.60	49.17	40.13	22.30	43.20	**26.23**
	D-ReLU	**82.90**	**61.03**	**56.40**	**27.90**	**51.10**	18.33
InceptionV3	TRADES-1	82.53	64.17	60.63	34.50	**60.43**	**39.60**
	TRADES-6	76.97	62.40	51.10	34.03	50.43	36.10
	D-ReLU	**87.17**	**74.20**	**67.07**	**41.40**	42.63	24.63

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 in a large dataset like the TinyImagenet dataset.

Overall, the findings underscore the robustness of the D-ReLU function against the black-box attack, 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.

5.4 Experiments with Augmented Dataset

The study conducted by Wang et al. (2023), 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. (2022) as a means to effectively mitigate the prevalent issue of overfitting encountered during adversarial training processes. By augmenting the training dataset with 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. (2019). 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. (2023), 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), 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.

5.4.1 Experimental Results

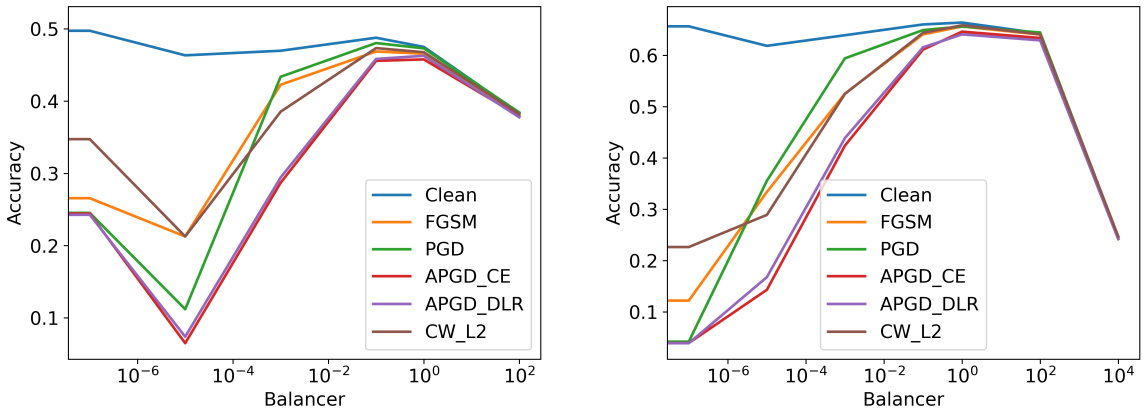
The visual representations displayed in Figure 5.13 for CIFAR10 and Figure 5.14 for CIFAR100 offer an insightful analysis into the performance and robustness under whitebox attacks of various architectures trained with D-ReLU, leveraging a training dataset enriched with generated samples from EDM. The fusion of D-ReLU with EDM showcases impressive results on both 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 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.

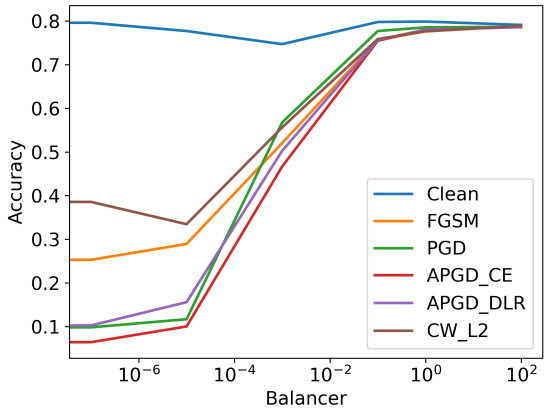
Tables 5.6 and 5.7 provide a comparative analysis between our approach using D-ReLU and the TRADES method with genereated samples from EDM across the CIFAR10 and CIFAR100 datasets respectively. Also, Tables A.5 and A.6 show the rankings for comparison. provide a comparative analysis between our approach using D-ReLU and the TRADES method with genereated samples from EDM across the CIFAR10 and CIFAR100 datasets respectively. 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 a 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.

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 models employed. 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.

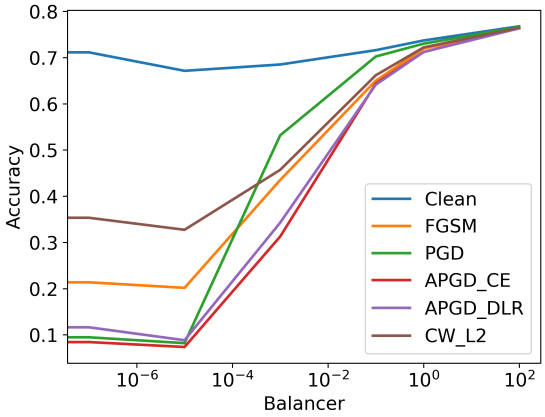


(a) Dense

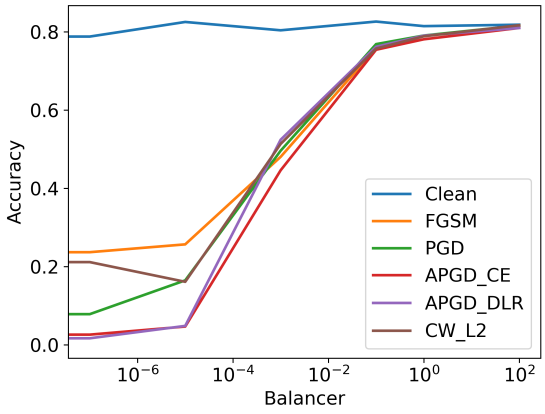
(b) Shallow CNN



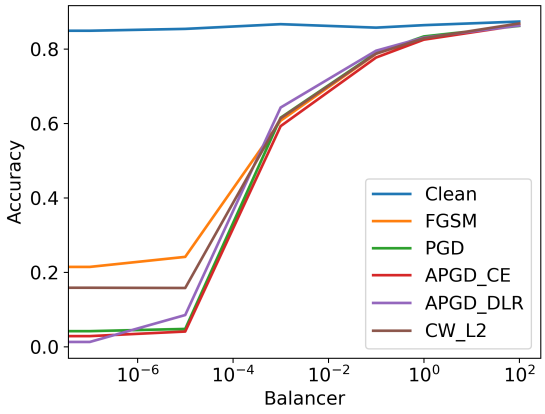
(c) ResNet50



(d) ResNet101

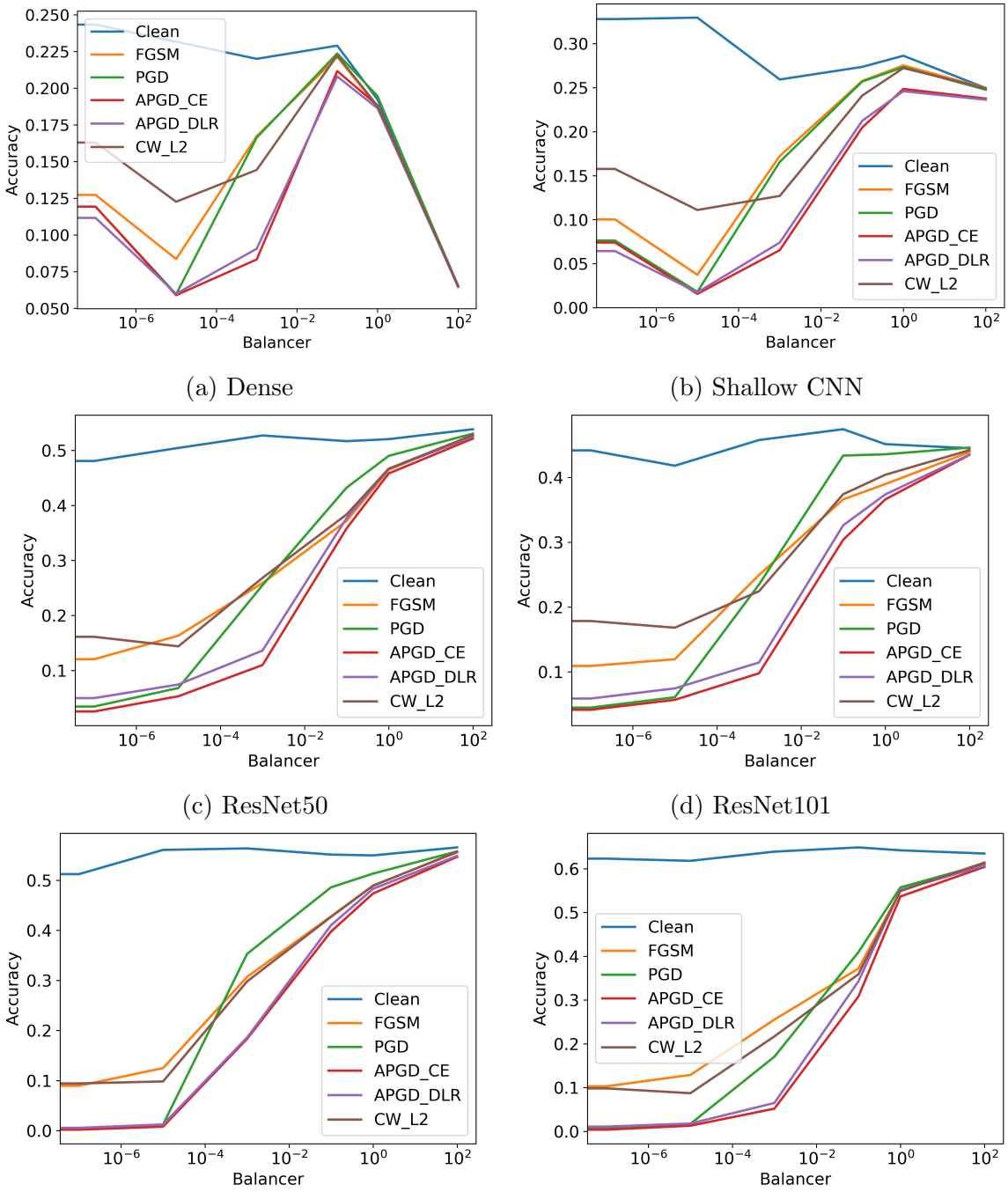


(e) MobilenetV2



(f) InceptionV3

Figure 5.13. Accuracy of several types of networks on clean CIFAR10 and adversarial examples when adding the dense layer with a D-ReLU function before the output layer and training them with augmented data samples generated from EDM.



(a) Dense

(b) Shallow CNN

(c) ResNet50

(d) ResNet101

(e) MobilenetV2

(f) InceptionV3

Figure 5.14. Accuracy of several types of networks on clean CIFAR100 and adversarial examples when adding the dense layer with a D-ReLU function before the output layer and training them with augmented data samples generated from EDM.

Table 5.6. Accuracy metrics for multiple types of networks under various robust training schemes with generated samples from EDM, evaluating on both clean samples and adversarial examples generated by different whitebox attacks on the CIFAR10 dataset. Note that the accuracy metrics in bold are the highest in a specific model among the different training methods.

Model	Training	Clean %	FGSM %	PGD %	APGD_{CE} %	APGD_{DLR} %	CW_{L2} %
Dense	D-ReLU	48.47	**46.87**	**48.03**	45.57	**45.83**	47.33
	TRADES	**62.47**	46.67	46.07	**46.13**	44.63	**52.8**
Shallow	D-ReLU	67.97	**66.57**	**67.07**	**65.4**	**65.4**	**66.97**
CNN	TRADES	**74.3**	59.03	57.93	57.93	56.53	**63.6**
Resnet	D-ReLU	79.1	**78.87**	**78.67**	**78.63**	**78.57**	**78.87**
50	TRADES	**80.6**	66.77	65.97	65.5	64.03	70.2
Resnet	D-ReLU	76.77	**76.37**	**76.63**	**76.43**	**76.33**	**76.43**
101	TRADES	**77.97**	63.43	61.93	61.87	59.77	67.33
Mobilenet	D-ReLU	**81.8**	**81.47**	**81.6**	**80.97**	**80.**97	**81.67**
V2	TRADES	79.33	62.27	61.1	60.67	58.4	66.87
Inception	D-ReLU	87.4	**86.77**	**86.23**	**86.4**	**86.33**	**86.9**
V3	TRADES	**87.73**	74.53	73.17	73.07	72.1	75.93

Table 5.7. Accuracy metrics for multiple types of networks under various robust training schemes with generated samples from EDM, evaluating on both clean samples and adversarial examples generated by different whitebox attacks on the CIFAR100 dataset. Note that the accuracy metrics in bold are the highest in a specific model among the different training methods.

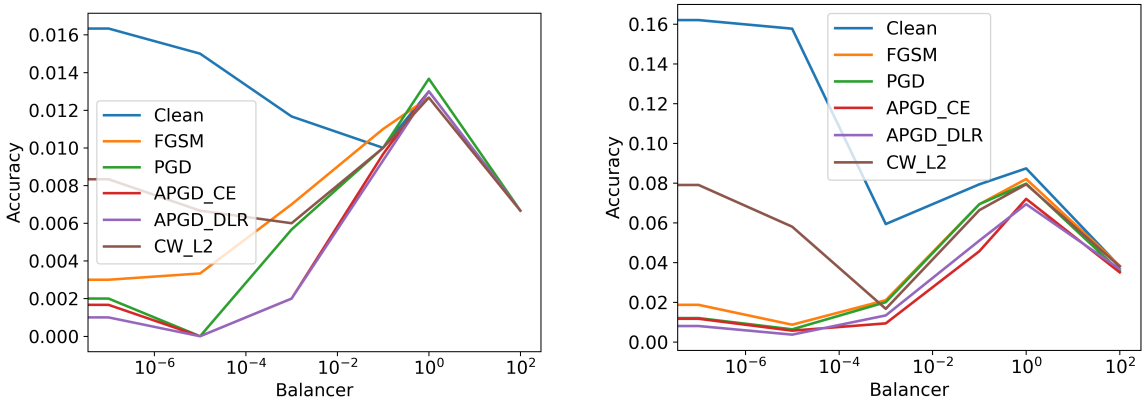
Model	Training	Clean %	FGSM %	PGD %	APGD_{CE} %	APGD_{DLR} %	CW_{L2} %
Dense	D-ReLU	22.9	22.13	22.37	21.17	20.8	22.23
	TRADES	**36.03**	**23.93**	**23.57**	**23.47**	**22.13**	**26.97**
Shallow	D-ReLU	32.2	**31.5**	**31.7**	**28.57**	**28.5**	**31.03**
CNN	TRADES	**44.23**	29.9	29.33	29.3	26.93	33.9
Resnet	D-ReLU	53.83	**52.8**	**53.03**	**52.13**	**52.5**	**52.77**
50	TRADES	**55.33**	40.17	38.03	37.8	37.27	43.13
Resnet	D-ReLU	44.5	**43.9**	**44.6**	**43.47**	**43.5**	**44.2**
101	TRADES	**52.6**	37.73	36.23	36.03	34.57	41.27
Mobilenet	D-ReLU	**56.57**	**55.57**	**55.77**	**54.67**	**54.87**	**55.7**
V2	TRADES	51.27	38.57	37.1	36.73	35.5	40.9
Inception	D-ReLU	**63.47**	**61.43**	**61.07**	**60.4**	**60.7**	**61.33**
V3	TRADES	62.9	48.33	46.5	46.23	45.67	49.43

The graphical representation provided in Figure 5.15 presents a detailed evaluation of the outcomes derived from implementing D-ReLU in conjunction with EDM on the TinyImagenet dataset. Interestingly, the results indicate a noticeable discrepancy in both performance and robustness compared to scenarios where solely D-ReLU is deployed. This inferior performance observed in the D-ReLU combined with EDM approach can be attributed to a crucial factor: the generated samples utilized for augmentation originate from data points that are external to the test dataset.

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.

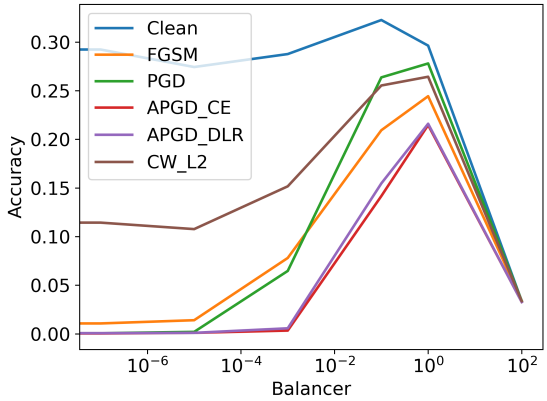
Table 5.8 and A.7 present a detailed comparison of the D-ReLU and TRADES training methodologies using samples generated from the EDM approach, particularly within the context of the TinyImagenet dataset. Upon examining the results, it becomes noticeable that the performance of D-ReLU in smaller network 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.

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

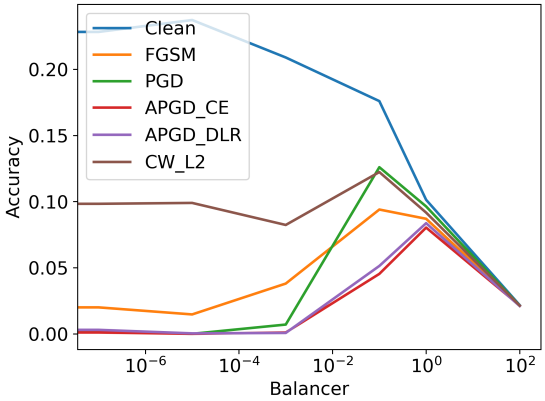


(a) Dense

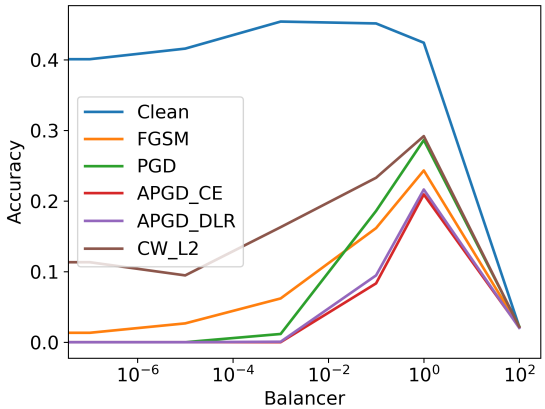
(b) Shallow CNN



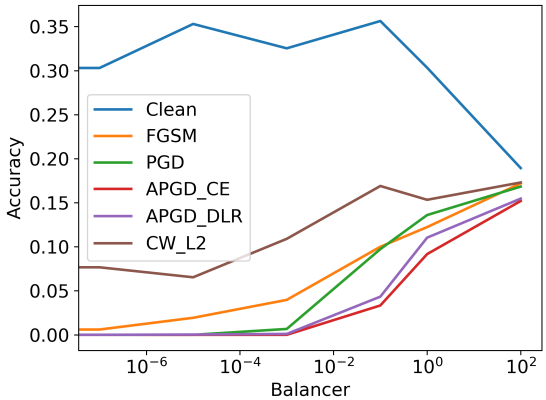
(c) ResNet50



(d) ResNet101



(e) MobilenetV2



(f) InceptionV3

Figure 5.15. Accuracy of several types of networks on clean TinyImagenet and adversarial examples when adding the dense layer with a D-ReLU function before the output layer and training them with augmented data samples generated from EDM.

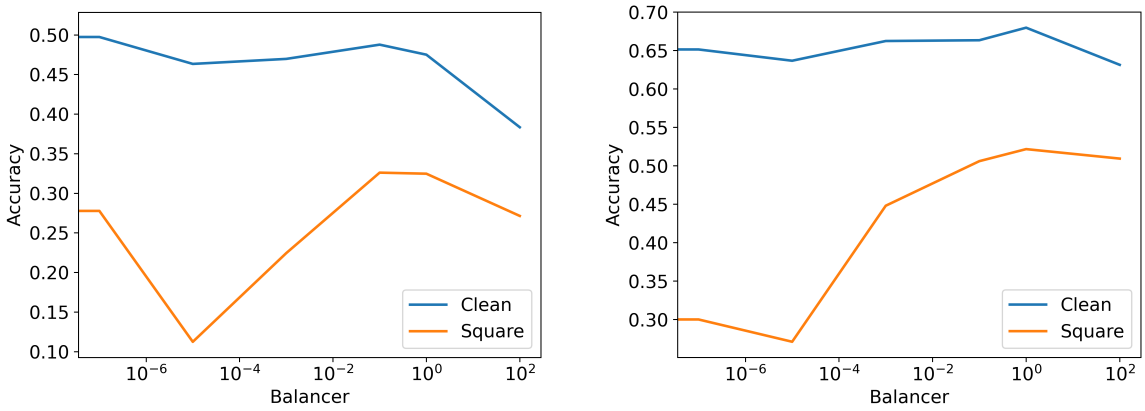
Table 5.8. Accuracy metrics for multiple types of networks under various robust training schemes with generated samples from EDM, evaluating on both clean samples and adversarial examples generated by different whitebox attacks on the TinyImagenet dataset. Note that the accuracy metrics in bold are the highest in a specific model among the different training methods.

Model	Training	Clean %	FGSM %	PGD %	APGD_{CE} %	APGD_{DLR} %	CW_{L2} %
Dense	D-ReLU	1.3	**1.27**	**1.37**	**1.3**	**1.3**	1.27
	TRADES	**2.4**	1.07	1.07	1.03	0.8	**1.77**
Shallow	D-ReLU	1.87	1.77	1.77	1.5	**1.53**	1.8
CNN	TRADES	**7.33**	**1.97**	**1.87**	**1.87**	1.13	**4.6**
Resnet	D-ReLU	**29.63**	**24.43**	**27.8**	**21.47**	**21.6**	**26.43**
50	TRADES	8.63	4.13	3.7	3.57	2.9	5.97
Resnet	D-ReLU	**17.6**	**9.4**	**12.6**	**4.53**	**5.13**	**12.23**
101	TRADES	7.3	3.63	3.37	3.33	2.87	5.2
Mobilenet	D-ReLU	**42.43**	**24.43**	**28.63**	**20.93**	**21.63**	**29.2**
V2	TRADES	18.13	8	7.03	6.63	5.2	12.63
Inception	D-ReLU	**35.63**	**10**	**9.73**	3.33	**4.33**	**16.9**
V3	TRADES	12.2	5.57	5.07	**5**	4.3	7.63

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.

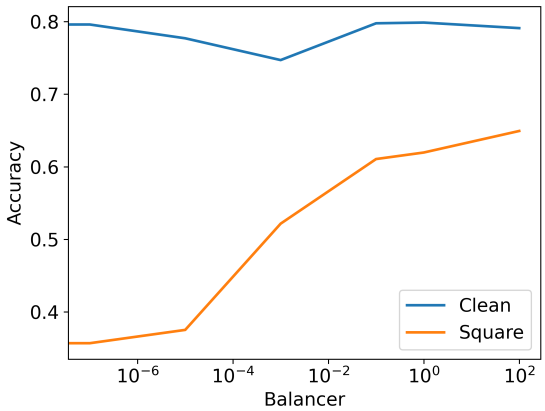
Figure 5.16, 5.17 and 5.18 visualize the accuracy on the clean and adversarial samples under several architectures on the CIFAR10, CIFAR100 and TinyImagenet datasets. These results follow the same patterns as in the whitebox attacks.

Table 5.9 and A.8 present a comparative analysis between the D-ReLU and TRADES methodologies, utilizing samples generated from the EDM approach while assessing the performance under a blackbox 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 from a blackbox attack align closely with those obtained from whitebox attacks, indicating a consistent behavior across different types of adversarial attacks in these simpler

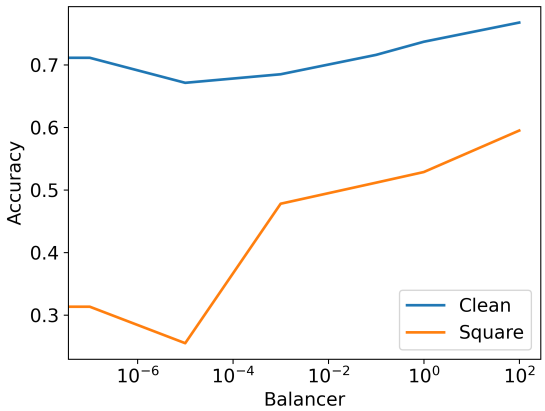


(a) Dense

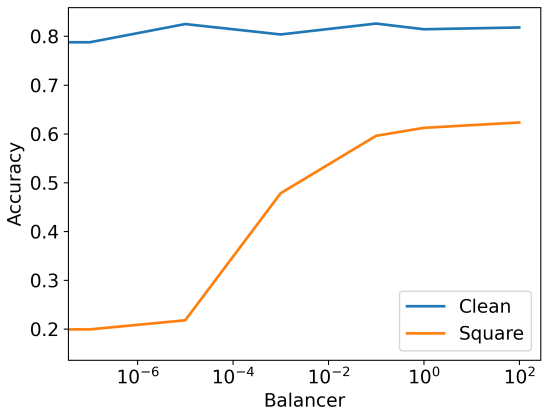
(b) Shallow CNN



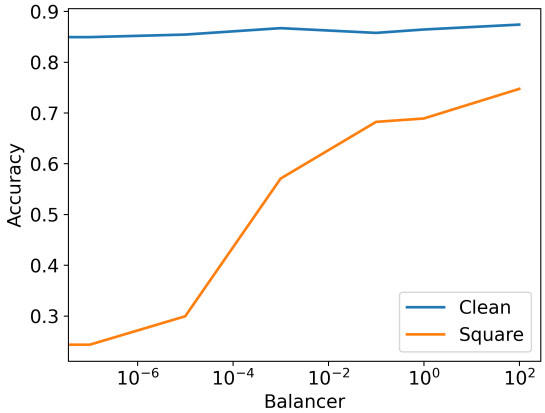
(c) ResNet50



(d) ResNet101

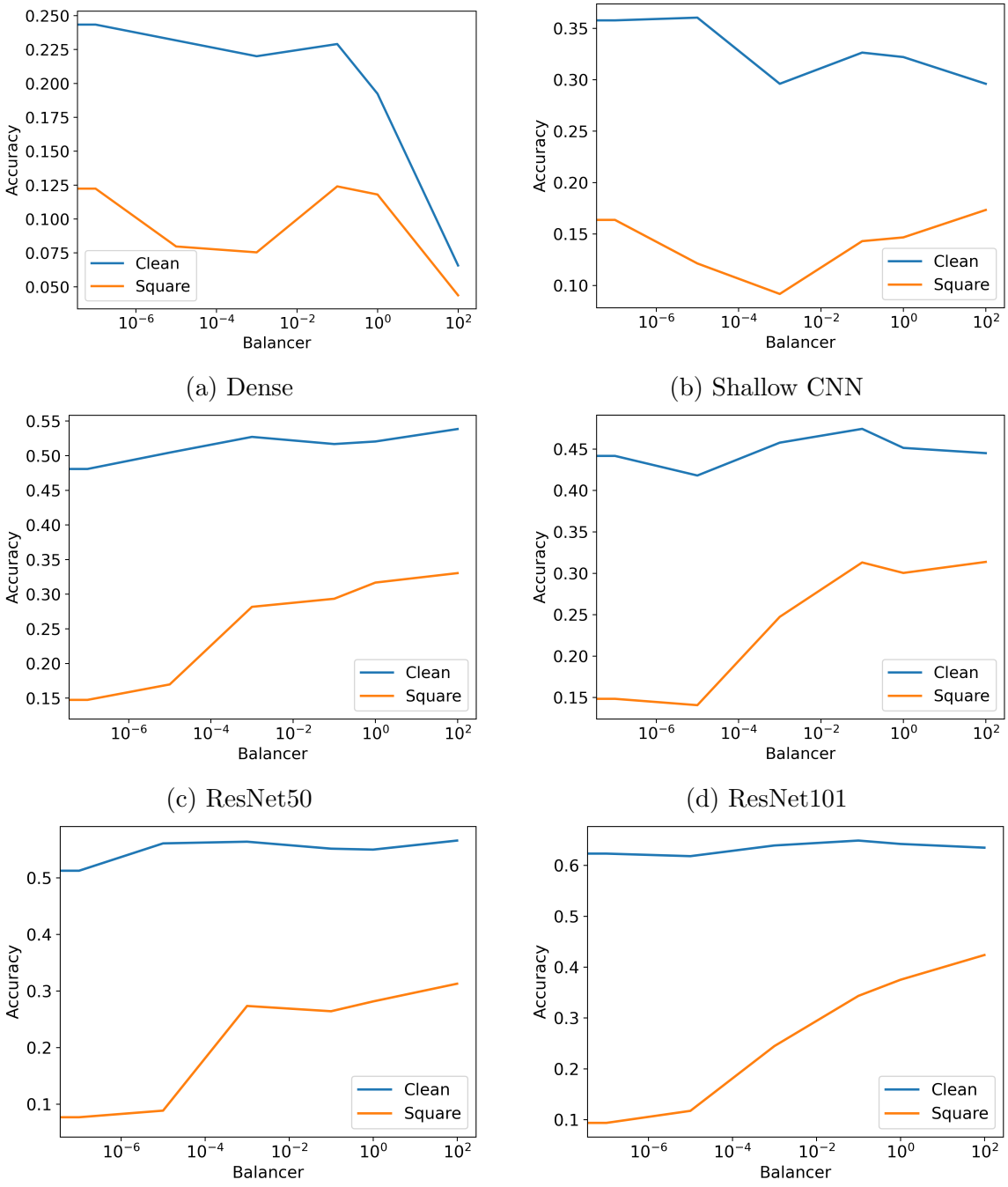


(e) MobilenetV2



(f) InceptionV3

Figure 5.16. Accuracy of several types of networks on clean CIFAR10 and adversarial examples generated by a blackbox attack (i.e., square attack) when adding the dense layer with a D-ReLU function before the output layer and training them with augmented data samples generated from EDM.



(a) Dense

(b) Shallow CNN

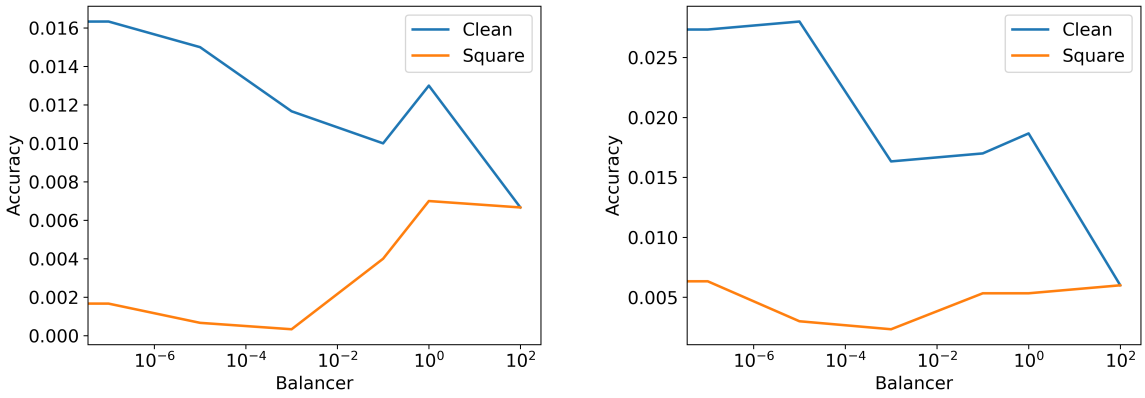
(c) ResNet50

(d) ResNet101

(e) MobilenetV2

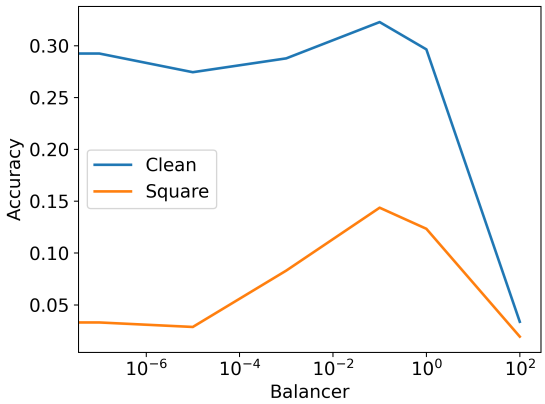
(f) InceptionV3

Figure 5.17. Accuracy of several types of networks on clean CIFAR100 and adversarial examples generated by a blackbox attack (i.e., square attack) when adding the dense layer with a D-ReLU function before the output layer and training them with augmented data samples generated from EDM.

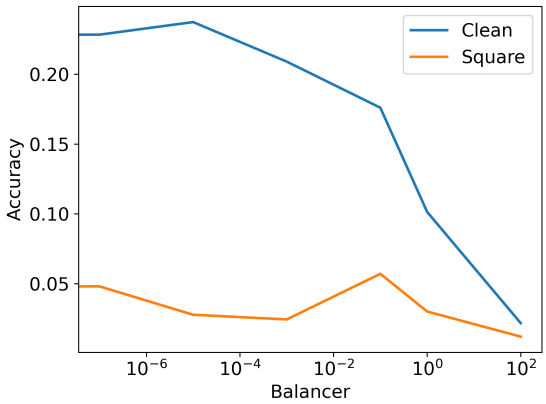


(a) Dense

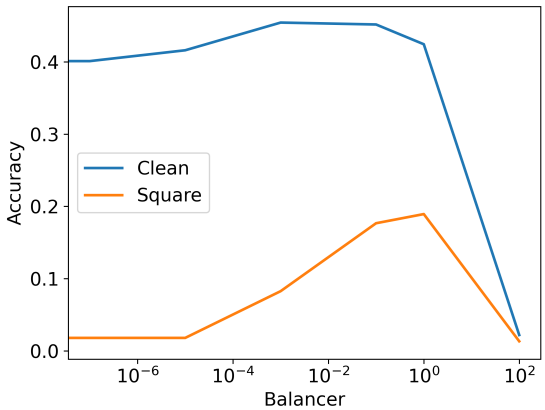
(b) Shallow CNN



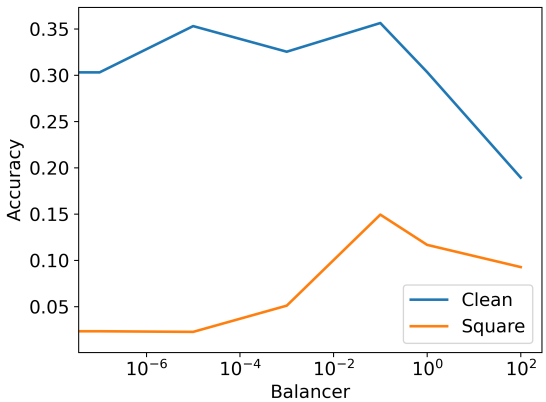
(c) ResNet50



(d) ResNet101



(e) MobilenetV2



(f) InceptionV3

Figure 5.18. Accuracy of several types of networks on clean TinyImagenet and adversarial examples generated by blackbox attacks when adding the dense layer with a D-ReLU function before the output layer and training them with augmented data samples generated from EDM.

network models. This consistency is crucial for validating the robustness of training methodologies against varied adversarial strategies.

Table 5.9. Accuracy for multiple types of networks under various robust training schemes with generated samples from EDM, evaluating on both clean samples and adversarial examples generated by a blackbox 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.

Model	Training	CIFAR10		CIFAR100		TinyImagenet	
		Clean %	Square %	Clean %	Square %	Clean %	Square %
Dense	D-ReLU	52.6	**48.77**	22.9	12.4	1.3	0.7
	TRADES	**62.47**	47.23	**36.03**	**23.6**	**2.4**	**0.93**
Shallow CNN	D-ReLU	67.97	52.17	35.3	14.43	2.67	0.5
	TRADES	**74.3**	**60.9**	**44.23**	**31.2**	**7.33**	**3.13**
Resnet 50	D-ReLU	79.1	64.93	53.83	33.03	**32.27**	**14.37**
	TRADES	**80.6**	**67.9**	**55.33**	**40.07**	7.33	4.7
Resnet 101	D-ReLU	76.77	59.5	47.43	31.3	**17.6**	**5.7**
	TRADES	**77.97**	**64.53**	**52.6**	**37.97**	7.3	3.87
Mobilenet V2	D-ReLU	**81.8**	62.33	**56.57**	31.27	**42.43**	**18.93**
	TRADES	79.33	**64.53**	51.27	**37.97**	18.13	9.9
Inception V3	D-ReLU	87.4	74.73	**63.47**	42.37	**35.63**	**14.93**
	TRADES	**87.73**	**76.63**	62.9	**48.8**	12.2	6.63

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 blackbox 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 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.

5.4.2 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 EDM.

However, the scenario shifts quite dramatically when considering the Tiny-Imagenet 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 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 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.

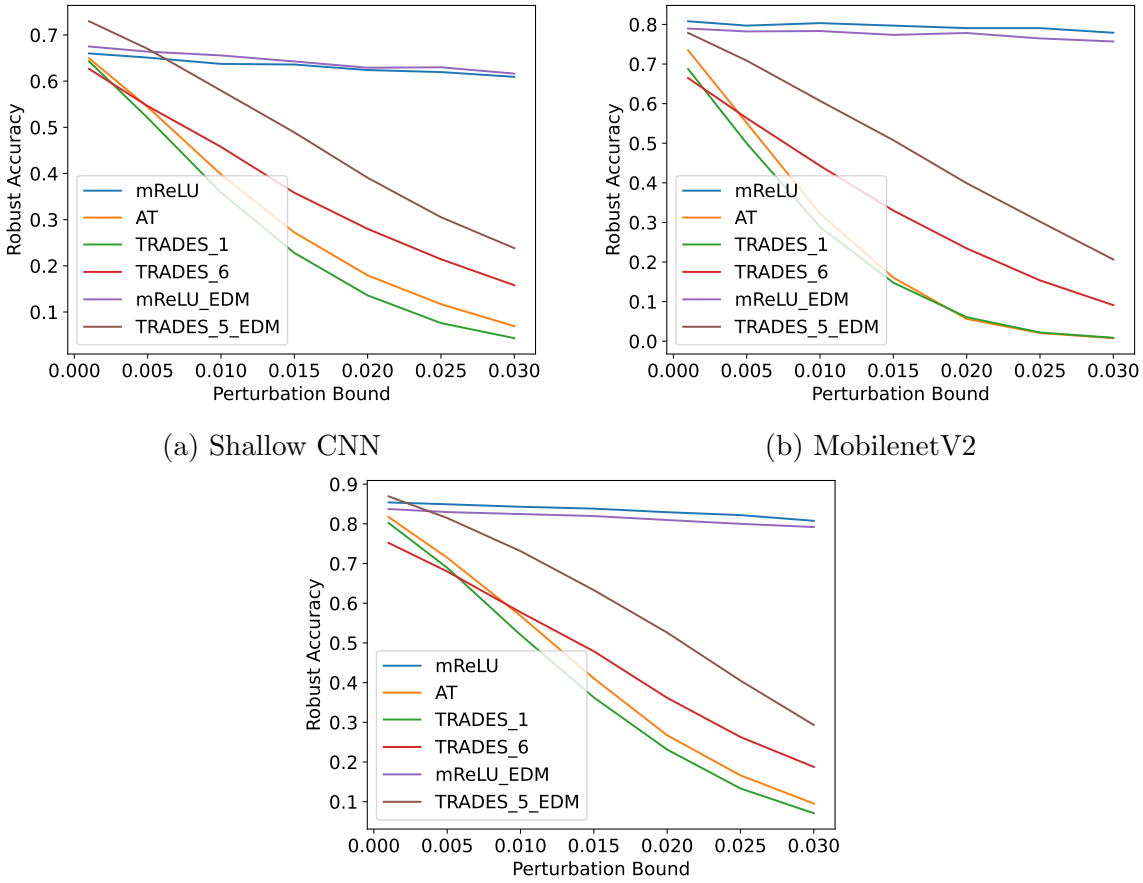
Such findings underscore the importance of contextual suitability of data augmentation techniques like EDM in training robust machine learning models. While EDM proves advantageous in datasets like CIFAR10 and CIFAR100 by enhancing model generalization through diverse examples, its effectiveness is contingent 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 optimizing model performance and robustness. This nuanced approach to training can significantly influence the successful deployment of machine learning models across various real-world applications.

5.5 Perturbation Bound Generalization

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

5.5.1 Experimental Results

Figure 5.19 presents the accuracy of various approaches, including the baselines and our proposed methods, on the CIFAR10 dataset under the APGD_CE attack with different levels of perturbation. For a small network like Shallow CNN, our approaches, D-ReLU and D-ReLU with EDM, outperform the other baselines under very small perturbations, with the exception of TRADES-5 with EDM. However, as the perturbation level increases, D-ReLU and D-ReLU with EDM consistently surpass all the baselines, demonstrating their superior robustness.



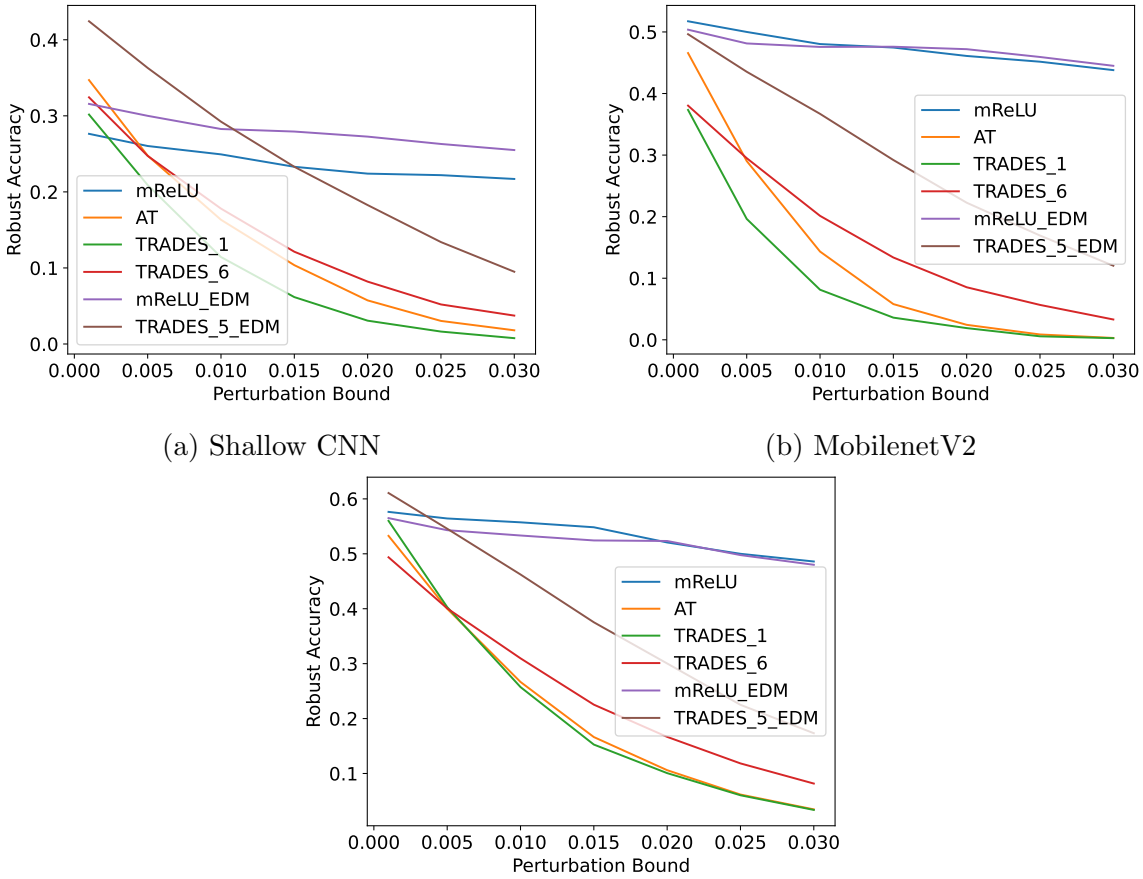
(a) Shallow CNN

(b) MobilenetV2

(c) InceptionV3

Figure 5.19. Accuracy of several approaches on the CIFAR10 dataset under the APGD_CE attack with various perturbation bounds where mReLU is D-ReLU

Figures 5.20 and 5.21 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 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

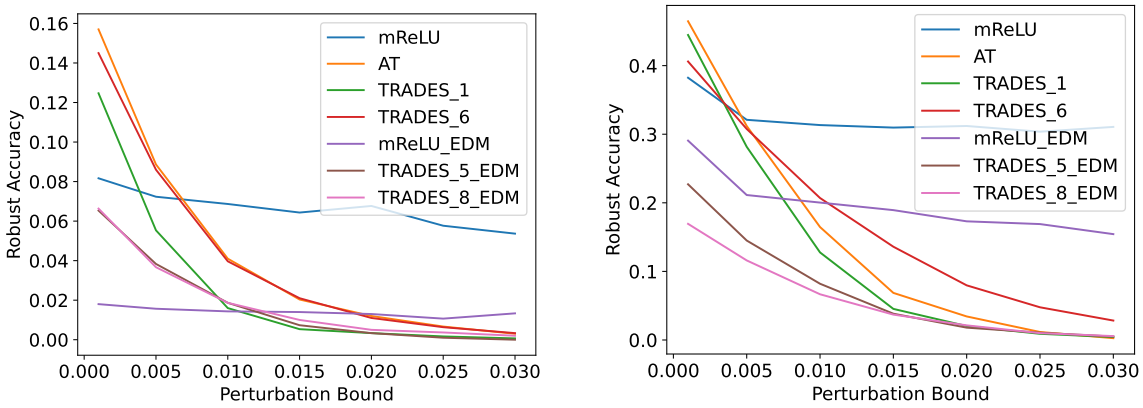
(b) MobilenetV2

(c) InceptionV3

Figure 5.20. Accuracy of several approaches on the CIFAR100 dataset under the APGD_CE attack with various perturbation bounds where mReLU is D-ReLU.

5.5.2 Discussion

Our approaches, D-ReLU and D-ReLU with 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.



(a) Shallow CNN

(b) MobilenetV2

(c) InceptionV3

Figure 5.21. Accuracy of several approaches on the TinyImagenet dataset under the APGD_CE attack with various perturbation bounds where mReLU is D-ReLU.

5.6 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 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, and it causes 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 the initial max value of 100 are not very bad. It is noteworthy that if this value is ridiculously high, this training time will significantly increase because the optimizer takes much more time to reduce this max value.

5.7 Conclusion

In this chapter, 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 whitebox and blackbox attack scenarios. Also, our statistical tests on the CIFAR10 dataset show that D-ReLU significantly outperforms the other baselines.

Moreover, even when testing with augmented samples from 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.

CHAPTER SIX

Conclusion

In this final chapter, we provide a comprehensive summary and discussion of the conclusions derived from this dissertation. Throughout this chapter, we will meticulously outline various key aspects of the research, including its intellectual merit, broader impact, significant contributions, and potential avenues for future work.

6.1 Intellectual Merit

This research sought to make significant advancements in the field of adversarial machine learning by focusing on innovation at the architecture level, specifically through the customization of Rectified Linear Unit (ReLU) activation functions. Our work set out to rigorously explore and define the most effective methods for tailoring activation functions and determining the optimal layers for their application. The principal enhanced the robustness of models against adversarial attacks while maintaining, or even improving, their performance.

To achieve this, our research systematically investigated S-ReLU, assessing where these modifications should be applied within the neural network architecture to best counteract adversarial manipulations without degrading performance on non-adversarial inputs. We employ a comprehensive evaluation framework that tested the modified architectures with S-ReLU and D-ReLU against various sophisticated adversarial attacks, including the Fast Gradient Sign Method (FGSM), Projected Gradient Descent (PGD), Auto PGD (APGD) and the Carlini and Wagner Attack (C&W). The impact of these customizations was quantitatively analyzed using three critical metrics: standard accuracy, robust accuracy, and attack success rates.

Moreover, this research benchmarked the customized models against existing defense mechanisms such as adversarial training and data augmentation. Through

these comparative analyses, we illustrated the relative efficacy and practicality of our proposed approach in a real-world adversarial context.

To underpin the empirical findings, the research integrated a thorough theoretical analysis, providing robust mathematical proof of the enhanced security features. This theoretical documentation demonstrated how and why S-ReLU and D-ReLU led to increased resistance to adversarial interventions compared to conventional activation models.

This project intended not merely to adapt existing models for greater security but to pioneer a methodological shift in how machine learning architectures could be intrinsically designed for resilience against evolving adversarial threats. Through both rigorous empirical testing and theoretical grounding, this research contributed substantially to the robustness of machine learning systems, aiming to set a new standard in the field.

6.2 Broader Impact

Our research was 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 was a critical advantage for machine learning practitioners who needed to ensure that the pursuit of robustness did not come at the expense of efficiency and overall model accuracy.

The potential applications of this technology extended far beyond academic research; it had practical, real-world implications across various sectors utilizing artificial intelligence. Industries ranging from finance and healthcare to autonomous vehicle technology and cybersecurity greatly benefited from integrating our findings into their AI development cycles. By implementing our advanced techniques, these

sectors were able to enhance the reliability and security of their systems against adversarial attacks, thus safeguarding sensitive data and critical operational functions.

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

At last, the broader impacts of this research were 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 industry applications. This work supported the important goal of advancing technology that was both powerful and resistant to evolving threats, thereby fostering a safer and more reliable digital future.

6.2.1 Publications

This research has yielded significant results, evidenced by the production of a publication, which highlights the multitude of applications as well as theoretical advancements. Below, we outline the publications that have been produced and discuss potential future works. First, we list publications that are closely related to this dissertation here:

- "Is ReLU Adversarially Robust?", Status: Published at the LatinX AI Workshop at ICML in 2023.
- "Dynamic-Max-Value ReLU Functions for Improving Adversarial Robustness of Deep Learning Models", Status: Drafted for Transactions on Machine Learning Research (TMLR).

- "Adversarial Defenses for Convolutional Neural Networks in Image Classification Task: A Survey", Status: Drafted for Transactions on Machine Learning Research (TMLR).

Moreover, we have other peer-reviewed and published research that was produced on adjacent topics to this dissertation but strongly related to the focus area of adversarial robustness and safety:

- "Evaluating Accuracy and Adversarial Robustness of Quanvolutional Neural Networks", Status: Published at International Conference on Computational Science and Computational Intelligence (CSCI) in 2021.
- "Adversarial Training Negatively Affects Fairness", Status: Published at International Conference on Computational Science and Computational Intelligence (CSCI) in 2021.
- "Enhancing Adversarial Examples on Deep Q Networks with Previous Information", Status: Published at IEEE Symposium Series on Computational Intelligence (SSCI) in 2021.
- "Evaluation of Adversarial Attacks Sensitivity of Classifiers with Occluded Input Data:, Status: Published at Neural Computing and Applications in 2022.
- "On Adversarial Examples for Text Classification by Perturbing Latent Representations", Status: Published at the LatinX AI Workshop at NeurIPS in 2022.
- "Evaluating Robustness of Reconstruction Models with Adversarial Networks", Status: Published at Procedia Computer Science in 2023.

6.3 Contributions

This dissertation achieves the critical need for improved adversarial defenses in machine learning by thoroughly examining the "capping technique" applied to ReLU functions. Through a systematic exploration and application of modifications to this

essential activation function, this research aims to bridge the current gap in robust defensive strategies against adversarial attacks.

6.3.1 Development of S-ReLU

The first significant part of our investigation focuses on the development and analysis of S-ReLU. We begin with a detailed theoretical analysis of S-ReLU to demonstrate its capacity to mitigate the effects of adversarial perturbations more effectively than the traditional ReLU function. This theoretical foundation supports the premise that S-ReLU can provide better protection against adversarial interference within neural network models.

Following the theoretical groundwork, we embark on empirical testing to validate the robustness of the S-ReLU. Through rigorous experiments, we establish that S-ReLU achieves superior robustness compared to conventional ReLU functions. Further, we benchmark S-ReLU against current state-of-the-art defense mechanisms, including methods like adversarial training, to highlight its enhanced defensive capabilities. These comparisons are critical in positioning S-ReLU as a formidable strategy against adversarial attacks.

6.3.2 Development of D-ReLU

Building upon the successes of S-ReLU, we further refine this approach by developing D-ReLU. The modification involves setting the maximum values of S-ReLU to align with certain parameters in machine learning models and adjusting the loss function to minimize these maximum values. This adjustment is based on the insights garnered from the theoretical analysis of S-ReLU, aiming to enhance the generalizability of the function across larger datasets.

To empirically substantiate the effectiveness of D-ReLU, we conduct an array of experimental evaluations, including white-box attacks, black-box scenarios, as well as tests involving data augmentation and perturbation-bound generalization. These

extensive experiments are designed to demonstrate how D-ReLU not only stands its ground but significantly outperforms contemporary state-of-the-art techniques in various aspects.

Overall, the research embodied in this dissertation provides a comprehensive exploration and enhancement of ReLU capping techniques, introducing innovative defenses that significantly bolster the adversarial robustness of machine learning systems. By advancing the understanding and application of S-ReL and D-ReLU, this work contributes valuable methodologies to the field of adversarial machine learning, paving the way for more secure AI implementations.

6.4 Future Works

The scope of this study, while comprehensive, has illuminated numerous areas ripe for further exploration. As highlighted in the discussions of limitations in previous chapters, the utilization of Dynamic-ReLU (D-ReLU) introduces additional hyperparameters that could significantly affect the performance and efficacy of machine learning models. Among these, factors like the initial maximum value of D-ReLU before training and the balancer settings are pivotal.

In the course of this dissertation research, considerable attention was devoted to examining the role of the balancer parameter. Our findings indicate that a balancer value of 1 yields optimal results in a multitude of scenarios. This insight not only validates our initial hypotheses but also enhances our understanding of the dynamic interactions within the activation function under adversarial conditions.

However, one aspect that has not been thoroughly investigated is the impact of the initial maximum value of D-ReLU. This parameter represents a fundamental aspect of how the D-ReLU function initially interacts with the incoming data, possibly affecting the learning process and the model's ultimate performance. Recognizing this gap, future work will be directed at extensively exploring various settings of the initial max value.

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.

6.5 Acknowledgements

This work was partially supported by the National Science Foundation under Grant Nos. 2039678, 2136961, and 2210091. The views expressed herein are solely those of the author and do not necessarily reflect those of the National Science Foundation.

APPENDIX

APPENDIX A

Table A.1, A.2, A.3, A.4, A.5, A.6, A.7 and A.8 are the rankings corresponding to the following tables, respectively: Table 5.2, 5.3, 5.4, 5.5, 5.6, 5.7, 5.8 and 5.9.

Table A.1. Ranking based on the accuracy metrics for multiple types of networks under various robust training schemes, evaluating on both clean samples and adversarial examples generated by different attacks on the CIFAR10 dataset. Note that AP_{CE} is APGD_{CE}, AP_{DLR} is APGD_{DLR}, TRADES-k means the TRADES approach with $\beta = k$, and D-ReLU-k means the D-ReLU approach with $m = k$.

Model	Training	Clean	FGSM	PGD	AP_{CE}	AP_{DLR}	CW_{L2}	$\bar{\mu}$
Dense	AT	1	2	2	2	2	2	1.8
	TRADES-1	2	3	3	3	3	3	2.8
	TRADES-6	4	1	1	1	1	1	1.5
	D-ReLU-10^{-7}	3	4	4	4	4	4	3.8
Shallow CNN	AT	3	2	2	2	2	2	2.2
	TRADES-1	1	3	3	3	3	3	2.7
	TRADES-6	4	1	1	1	1	1	1.5
	D-ReLU-10^{0}	2	4	4	4	4	4	3.7
ResNet 50	AT	2	2	3	3	2	2	2.3
	TRADES-1	3	3	4	4	3	3	3.5
	TRADES-6	4	2	2	2	4	3	2.8
	D-ReLU-10^{4}	1	1	1	1	1	1	1.0
ResNet 101	AT	3	3	2	2	2	3	2.5
	TRADES-1	2	1	4	4	4	1	2.7
	TRADES-6	4	2	3	3	3	4	3.2
	D-ReLU-10^{4}	1	4	1	1	1	2	1.7
Mobilenet V2	AT	2	3	3	3	3	3	2.7
	TRADES-1	3	4	4	4	4	4	3.8
	TRADES-6	4	2	2	2	2	2	2.3
	D-ReLU-10^{2}	1	1	1	1	1	1	1.0
Inception V3	AT	2	3	2	2	2	2	2.2
	TRADES-1	4	4	4	4	4	4	4.0
	TRADES-6	3	4	3	3	3	3	3.2
	D-ReLU-10^{2}	1	1	1	1	1	1	1.0

Table A.2. Ranking based on the accuracy metrics for multiple types of networks under various robust training schemes, evaluating on both clean samples and adversarial examples generated by different 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, TRADES-k means the TRADES approach with $\beta = k$, and D-ReLU-k means the D-ReLU approach with $m = k$.

Model	Training	Clean	FGSM	PGD	AP_{CE}	AP_{DLR}	CW_{L2}	$\bar{\mu}$
Dense	AT	1	2	2	2	2	2	1.8
	TRADES-1	2	4	4	4	4	4	3.7
	TRADES-6	3	3	3	3	3	3	3.0
	D-ReLU-10^{-1}	4	1	1	1	1	1	1.5
Shallow CNN	AT	1	2	2	2	2	2	1.8
	TRADES-1	3	4	4	4	4	4	3.8
	TRADES-6	2	3	3	3	3	3	2.8
	D-ReLU-1	4	1	1	1	1	1	1.5
ResNet 50	AT	3	3	2	2	3	3	2.7
	TRADES-1	2	4	4	4	4	4	3.7
	TRADES-6	4	2	3	3	2	2	2.7
	D-ReLU-10^2	1	1	1	1	1	1	1.0
ResNet 101	AT	3	3	2	2	2	3	2.5
	TRADES-1	4	4	4	4	4	2	3.7
	TRADES-6	2	2	3	3	3	3	2.7
	D-ReLU-1	1	1	1	1	1	1	1.0
Mobilenet V2	AT	2	2	3	3	3	2	2.5
	TRADES-1	3	3	4	4	4	4	3.7
	TRADES-6	4	4	2	2	2	3	2.8
	D-ReLU-1	1	1	1	1	1	1	1.0
Inception V3	AT	3	3	3	3	2	3	2.83
	TRADES-1	2	2	4	4	4	2	3.0
	TRADES-6	4	4	2	2	3	4	3.2
	D-ReLU-10^2	1	1	1	1	1	1	1.0

Table A.3. Ranking based on the accuracy metrics for multiple types of networks under various robust training schemes, evaluating on both clean samples and adversarial examples generated by different 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, TRADES-k means the TRADES approach with $\beta = k$, and D-ReLU-k means the D-ReLU approach with $m = k$.

Model	Training	Clean	FGSM	PGD	AP_{CE}	AP_{DLR}	CW_{L2}	$\bar{\mu}$
Dense	AT	2	3	2	2	3	2	2.3
	TRADES-1	3	4	3	3	4	4	3.5
	TRADES-6	1	2	2	2	2	3	2.0
	D-ReLU-10^{-1}	4	1	1	1	1	1	1.5
Shallow CNN	AT	1	3	3	3	3	2	2.5
	TRADES-1	2	4	4	4	4	4	3.7
	TRADES-6	3	2	2	2	2	1	2.0
	D-ReLU-1	4	1	1	1	1	3	1.8
ResNet 50	AT	3	3	4	4	3	3	3.3
	TRADES-1	1	2	3	3	2	1	2.0
	TRADES-6	2	4	2	2	4	2	2.7
	D-ReLU-1	4	1	1	1	1	4	2.0
ResNet 101	AT	3	4	4	4	4	4	3.8
	TRADES-1	1	3	3	3	2	1	2.2
	TRADES-6	2	1	2	1	3	2	1.8
	D-ReLU-1	4	2	1	2	1	3	2.2
Mobilenet V2	AT	2	3	3	3	3	2	2.7
	TRADES-1	3	4	4	4	4	4	3.8
	TRADES-6	4	2	2	2	2	3	2.5
	D-ReLU-1	1	1	1	1	1	1	1.0
Inception V3	AT	4	4	4	4	4	4	4.0
	TRADES-1	1	1	3	3	1	1	1.7
	TRADES-6	3	2	1	1	2	2	1.8
	D-ReLU-1	2	3	2	2	3	3	2.5

Table A.4. Ranking based on the accuracy metrics for multiple types of networks under various robust training schemes, evaluating on both clean samples and adversarial examples generated by a blackbox attach (i.e. Square) on the CIFAR10, CIFAR100 and TinyImagenet datasets. Note that Sq means the square attack, the accuracy metrics in bold are the highest in a specific model among the different training methods. Note that TRADES-k means the TRADES approach with $\beta = k$.

Model	Training	CIFAR10		CIFAR100		TinyImagenet		$\bar{\mu}$
		Clean	Sq	Clean	Sq	Clean	Sq	
Dense	TRADES-1	1	2	2	2	2	2	1.8
	TRADES-6	2	1	1	1	1	1	1.2
	D-ReLU	3	3	3	3	3	3	3.0
Shallow CNN	TRADES-1	2	2	2	2	2	2	2.0
	TRADES-6	3	1	1	1	1	1	1.3
	D-ReLU	1	3	3	3	3	3	2.7
Resnet50	TRADES-1	2	2	2	2	1	1	1.7
	TRADES-6	3	1	3	1	2	2	2.0
	D-ReLU	1	3	1	3	3	3	2.3
Resnet101	TRADES-1	1	2	1	2	1	1	1.3
	TRADES-6	3	3	3	3	2	2	2.7
	D-ReLU	2	1	2	1	3	3	2.0
MobilenetV2	TRADES-1	2	3	2	3	1	2	2.2
	TRADES-6	3	2	3	2	2	1	2.2
	D-ReLU	1	1	1	1	3	3	1.7
InceptionV3	TRADES-1	2	2	1	2	1	1	1.5
	TRADES-6	3	3	3	3	2	2	2.7
	D-ReLU	1	1	2	1	3	3	1.8

Table A.5. Ranking based on the accuracy metrics for multiple types of networks under various robust training schemes with generated samples from EDM, evaluating on both clean samples and adversarial examples generated by different whitebox 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.

Model	Training	Clean	FGSM	PGD	AP_{CE}	AP_{DLR}	CW_{L2}	$\bar{\mu}$
Dense	D-ReLU	2	1	1	2	1	2	1.5
	TRADES	1	2	2	1	2	1	1.5
Shallow	D-ReLU	2	1	1	1	1	1	1.2
CNN	TRADES	1	2	2	2	2	2	1.8
Resnet	D-ReLU	2	1	1	1	1	1	1.2
50	TRADES	1	2	2	2	2	2	1.8
Resnet	D-ReLU	2	1	1	1	1	1	1.2
101	TRADES	1	2	2	2	2	2	1.8
Mobilenet	D-ReLU	1	1	1	1	1	1	1.0
V2	TRADES	2	2	2	2	2	2	2.0
Inception	D-ReLU	2	1	1	1	1	1	1.2
V3	TRADES	1	2	2	2	2	2	1.8

Table A.6. Ranking based on the accuracy metrics for multiple types of networks under various robust training schemes with generated samples from EDM, evaluating on both clean samples and adversarial examples generated by different whitebox 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.

Model	Training	Clean	FGSM	PGD	AP_{CE}	AP_{DLR}	CW_{L2}	$\bar{\mu}$
Dense	D-ReLU	2	2	2	2	2	2	2.0
	TRADES	1	1	1	1	1	1	1.0
Shallow	D-ReLU	2	1	1	1	1	1	1.2
CNN	TRADES	1	2	2	2	2	2	1.8
Resnet	D-ReLU	2	1	1	1	1	1	1.2
50	TRADES	1	2	2	2	2	2	1.8
Resnet	D-ReLU	2	1	1	1	1	1	1.2
101	TRADES	1	2	2	2	2	2	1.8
Mobilenet	D-ReLU	1	1	1	1	1	1	1.0
V2	TRADES	2	2	2	2	2	2	2.0
Inception	D-ReLU	1	1	1	1	1	1	1.0
V3	TRADES	2	2	2	2	2	2	2.0

Table A.7. Ranking based on the accuracy metrics for multiple types of networks under various robust training schemes with generated samples from EDM, evaluating on both clean samples and adversarial examples generated by different whitebox 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.

Model	Training	Clean	FGSM	PGD	AP_{CE}	AP_{DLR}	CW_{L2}	$\bar{\mu}$
Dense	D-ReLU	2	1	1	1	1	2	1.3
	TRADES	1	2	2	2	2	1	1.7
Shallow	D-ReLU	2	2	2	2	1	2	1.8
CNN	TRADES	1	1	1	1	2	1	1.2
ResNet	D-ReLU	1	1	1	1	1	1	1.0
50	TRADES	2	2	2	2	2	2	2.0
ResNet	D-ReLU	1	1	1	1	1	1	1.0
101	TRADES	2	2	2	2	2	2	2.0
Mobilenet	D-ReLU	1	1	1	1	1	1	1.0
V2	TRADES	2	2	2	2	2	2	2.0
Inception	D-ReLU	1	1	1	2	1	1	1.2
V3	TRADES	2	2	2	1	2	2	1.8

Table A.8. Ranking based on the accuracy metrics for multiple types of networks under various robust training schemes with generated samples from EDM, evaluating on both clean samples and adversarial examples generated by a blackbox 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.

Model	Training	CIFAR10		CIFAR100		TinyImagenet		$\bar{\mu}$
		Clean	Square	Clean	Square	Clean	Square	
Dense	D-ReLU	2	1	2	2	1	1	1.5
	TRADES	1	2	1	1	2	2	1.5
Shallow	D-ReLU	2	2	2	2	2	2	2.0
CNN	TRADES	1	1	1	1	1	1	1.0
Resnet	D-ReLU	2	2	2	2	1	1	1.7
50	TRADES	1	1	1	1	2	2	1.3
Resnet	D-ReLU	2	2	2	2	1	1	1.7
101	TRADES	1	1	1	1	2	2	1.3
Mobilenet	D-ReLU	1	2	1	2	1	1	1.3
V2	TRADES	2	1	2	1	2	2	1.7
Inception	D-ReLU	2	2	1	2	1	1	1.5
V3	TRADES	1	1	2	1	2	2	1.5

BIBLIOGRAPHY

Alayrac, J.-B., J. Uesato, P.-S. Huang, A. Fawzi, R. Stanforth, and P. Kohli (2019). Are labels required for improving adversarial robustness? In *Advances in Neural Information Processing Systems*, Volume 32.

Alfarra, M., J. C. Pérez, A. Thabet, A. Bibi, P. H. Torr, and B. Ghanem (2022). Combating adversaries with anti-adversaries. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Volume 36, pp. 5992–6000.

Ali Mousavi, S., H. Mousavi, and M. Daneshtalab (2023). Farmur: Fair adversarial retraining to mitigate unfairness in robustness. In *European Conference on Advances in Databases and Information Systems*, pp. 133–145. Springer.

Amich, A. and B. Eshete (2021). Morphence: Moving target defense against adversarial examples. In *Annual Computer Security Applications Conference*, pp. 61–75.

Andriushchenko, M., F. Croce, N. Flammarion, and M. Hein (2020). Square attack: a query-efficient black-box adversarial attack via random search. In *European conference on computer vision*, pp. 484–501. Springer.

Atsague, M., A. Nirala, O. Fakorede, and J. Tian (2023). A penalized modified huber regularization to improve adversarial robustness. In *2023 IEEE International Conference on Image Processing (ICIP)*, pp. 2675–2679. IEEE.

Bai, T., J. Luo, J. Zhao, B. Wen, and Q. Wang (2021). Recent advances in adversarial training for adversarial robustness. In *arXiv preprint arXiv:2102.01356*.

Bai, Y., B. G. Anderson, A. Kim, and S. Sojoudi (2023). Improving the accuracy-robustness trade-off of classifiers via adaptive smoothing. In *arXiv preprint arXiv:2301.12554*.

Bai, Y., Y. Feng, Y. Wang, T. Dai, S.-T. Xia, and Y. Jiang (2019). Hilbert-based generative defense for adversarial examples. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4784–4793.

Bai, Y., M. Zhou, V. M. Patel, and S. Sojoudi (2024). Mixednuts: Training-free accuracy-robustness balance via nonlinearly mixed classifiers. In *arXiv preprint arXiv:2402.02263*.

Bubeck, S., Y. Li, and D. M. Nagaraj (2021). A law of robustness for two-layers neural networks. In *Conference on Learning Theory*, pp. 804–820. PMLR.

Bubeck, S. and M. Sellke (2021). A universal law of robustness via isoperimetry. *Advances in Neural Information Processing Systems 34*, 28811–28822.

Carlini, N. and D. Wagner (2016). Defensive distillation is not robust to adversarial examples. In *arXiv preprint arXiv:1607.04311*.

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

Chakraborty, A., M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay (2018). Adversarial attacks and defences: A survey. In *arXiv preprint arXiv:1810.00069*.

Chen, J., D. Yan, and L. Dong (2023). Adversarial defense based on distribution transfer. In *arXiv preprint arXiv:2311.13841*.

Chen, X., X. Li, Y. Zhou, and T. Yang (2022). Dddm: a brain-inspired framework for robust classification. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI-22)*.

Chitsaz, K., G. Mordido, J.-P. David, and F. Leduc-Primeau (2023). Training dnns resilient to adversarial and random bit-flips by learning quantization ranges. In *Transactions on Machine Learning Research*.

Chrabaszcz, P., I. Loshchilov, and F. Hutter (2017). A downsampled variant of imagenet as an alternative to the cifar datasets. In *arXiv preprint arXiv:1707.08819*.

Clanuwat, T., M. Bober-Irizar, A. Kitamoto, A. Lamb, K. Yamamoto, and D. Ha (2018). Deep learning for classical japanese literature.

Cohen, G., S. Afshar, J. Tapson, and A. Van Schaik (2017). Emnist: Extending mnist to handwritten letters. In *2017 international joint conference on neural networks (IJCNN)*, pp. 2921–2926. IEEE.

Cohen, J., E. Rosenfeld, and Z. Kolter (2019). Certified adversarial robustness via randomized smoothing. In *international conference on machine learning*, pp. 1310–1320. PMLR.

Croce, F. and M. Hein (2020). Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pp. 2206–2216. PMLR.

Cui, J., Z. Tian, Z. Zhong, X. Qi, B. Yu, and H. Zhang (2023). Decoupled kullback-leibler divergence loss. In *arXiv preprint arXiv:2305.13948*.

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

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine learning research 7*, 1–30.

Deng, L. (2012). The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine 29*(6), 141–142.

Doan, B. G., E. M. Abbasnejad, J. Q. Shi, and D. C. Ranasinghe (2022). Bayesian learning with information gain provably bounds risk for a robust adversarial defense. In *International Conference on Machine Learning*, pp. 5309–5323. PMLR.

Dong, J., S.-M. Moosavi-Dezfooli, J. Lai, and X. Xie (2023). The enemy of my enemy is my friend: Exploring inverse adversaries for improving adversarial training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24678–24687.

Dong, M. and C. Xu (2023). Adversarial robustness via random projection filters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4077–4086.

Gong, H., S. Yang, S. Ma, S. Camtepe, S. Nepal, and C. Xu (2023). Reducing adversarial training cost with gradient approximation. In *arXiv preprint arXiv:2309.09464*.

Goodfellow, I. J., J. Shlens, and C. Szegedy (2014). Explaining and harnessing adversarial examples. In *arXiv preprint arXiv:1412.6572*.

Gowal, S., C. Qin, J. Uesato, T. Mann, and P. Kohli (2020). Uncovering the limits of adversarial training against norm-bounded adversarial examples. In *arXiv preprint arXiv:2010.03593*.

Gowal, S., S.-A. Rebuffi, O. Wiles, F. Stimberg, D. A. Calian, and T. A. Mann (2021). Improving robustness using generated data. *Advances in Neural Information Processing Systems 34*, 4218–4233.

Guan, X., Q. Shao, Y. Qian, T. Yao, and B. Wang (2023). Adversarial training in logit space against tiny perturbations. *Multimedia Systems 29*(6), 3277–3290.

Gui, S., H. Wang, H. Yang, C. Yu, Z. Wang, and J. Liu (2019). Model compression with adversarial robustness: A unified optimization framework. In *Advances in Neural Information Processing Systems*, Volume 32.

He, K., X. Zhang, S. Ren, and J. Sun (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

He, L., Q. Ai, X. Yang, Y. Ren, Q. Wang, and Z. Xu (2023). Boosting adversarial robustness via self-paced adversarial training. *Neural Networks 167*, 706–714.

Huang, H., Y. Wang, S. Erfani, Q. Gu, J. Bailey, and X. Ma (2021). Exploring architectural ingredients of adversarially robust deep neural networks. *Advances in Neural Information Processing Systems 34*, 5545–5559.

Huang, J., Y. Dai, F. Lu, B. Wang, Z. Gu, B. Zhou, and Y. Qian (2024). Adversarial perturbation denoising utilizing common characteristics in deep feature space. In *Applied Intelligence*, pp. 1–19. Springer.

Huang, S., Z. Lu, K. Deb, and V. N. Boddeti (2023). Revisiting residual networks for adversarial robustness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8202–8211.

Iijima, R., S. Shiota, and H. Kiya (2024). A random ensemble of encrypted vision transformers for adversarially robust defense. In *arXiv preprint arXiv:2402.07183*.

Ilyas, A., L. Engstrom, A. Athalye, and J. Lin (2018). Black-box adversarial attacks with limited queries and information. In *International Conference on Machine Learning*, pp. 2137–2146. PMLR.

Jia, X., Y. Zhang, X. Wei, B. Wu, K. Ma, J. Wang, and X. Cao (2022). Prior-guided adversarial initialization for fast adversarial training. In *European Conference on Computer Vision*, pp. 567–584. Springer.

Kabilan, V. M., B. Morris, H.-P. Nguyen, and A. Nguyen (2021). Vectordefense: Vectorization as a defense to adversarial examples. In *Soft Computing for Biomedical Applications and Related Topics*, pp. 19–35. Springer.

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

Kang, Q., Y. Song, Q. Ding, and W. P. Tay (2021). Stable neural ode with lyapunov-stable equilibrium points for defending against adversarial attacks. *Advances in Neural Information Processing Systems 34*, 14925–14937.

Karras, T., M. Aittala, T. Aila, and S. Laine (2022). Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems 35*, 26565–26577.

Khan, S., J.-C. Chen, W.-H. Liao, and C.-S. Chen (2023). Towards adversarial robustness for multi-mode data through metric learning. *Sensors 23*(13), 6173.

Kingma, D. P. and J. Ba (2014). Adam: A method for stochastic optimization. In *arXiv preprint arXiv:1412.6980*.

Krizhevsky, A., G. Hinton, et al. (2009). Learning multiple layers of features from tiny images.

Kurakin, A., I. Goodfellow, S. Bengio, et al. (2016). Adversarial examples in the physical world.

Le, Y. and X. S. Yang (2015). Tiny imagenet visual recognition challenge.

LeCun, Y., C. Cortes, and C. Burges (2010). Mnist handwritten digit database. In *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, Volume 2.

Li, B., C. Chen, W. Wang, and L. Carin (2019). Certified adversarial robustness with additive noise. In *Advances in neural information processing systems*, Volume 32.

Li, J. and G. Li (2024, feb). The triangular trade-off between robustness, accuracy and fairness in deep neural networks: A survey. In *ACM Comput. Surv.*, New York, NY, USA. Association for Computing Machinery.

Li, L. and M. W. Spratling (2023). Data augmentation alone can improve adversarial training. In *The Eleventh International Conference on Learning Representations*.

Li, L., T. Xie, and B. Li (2023). Sok: Certified robustness for deep neural networks. In *2023 IEEE symposium on security and privacy (SP)*, pp. 1289–1310. IEEE.

Li, Q., J. Chen, K. He, Z. Zhang, R. Du, J. She, and X. Wang (2024). Model-agnostic adversarial example detection via high-frequency amplification. In *Computers & Security*, pp. 103791. Elsevier.

Li, Y., M. Cheng, C.-J. Hsieh, and T. C. Lee (2022). A review of adversarial attack and defense for classification methods. *The American Statistician 76*(4), 329–345.

Lin, G., C. Li, J. Zhang, T. Tanaka, and Q. Zhao (2023). Adversarial training on purification (atop): Advancing both robustness and generalization. In *The Twelfth International Conference on Learning Representations*.

Liu, J., W. Zhang, Y. Zhang, D. Hou, Y. Liu, H. Zha, and N. Yu (2019). 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*, pp. 4825–4834.

Liu, L., T. N. Hoang, L. M. Nguyen, and T.-W. Weng (2023). Promoting robustness of randomized smoothing: Two cost-effective approaches. In *2023 IEEE International Conference on Data Mining (ICDM)*, pp. 1145–1150. IEEE.

Lukasik, J., P. Gavrikov, J. Keuper, and M. Keuper (2023). Improving native cnn robustness with filter frequency regularization. In *Transactions on Machine Learning Research*.

Lyu, W., M. Wu, Z. Yin, and B. Luo (2023). Maedefense: An effective masked autoencoder defense against adversarial attacks. In *2023 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 1915–1922.

Ma, Y., M. Dong, and C. Xu (2024). Adversarial robustness through random weight sampling. In *Advances in Neural Information Processing Systems*, Volume 36.

Madry, A., A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu (2017). Towards deep learning models resistant to adversarial attacks. In *arXiv preprint arXiv:1706.06083*.

Mandal, S. (2023). Defense against adversarial attacks using convolutional auto-encoders. In *arXiv preprint arXiv:2312.03520*.

Meng, D. and H. Chen (2017). Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pp. 135–147.

Naseer, M., S. Khan, M. Hayat, F. S. Khan, and F. Porikli (2020). A self-supervised approach for adversarial robustness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 262–271.

Nesti, F., A. Biondi, and G. Buttazzo (2021). Detecting adversarial examples by input transformations, defense perturbations, and voting. In *IEEE Transactions on neural networks and learning systems*. IEEE.

Netzer, Y., T. Wang, A. Coates, A. Bissacco, B. Wu, A. Y. Ng, et al. (2011). Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, Volume 2011, pp. 7. Granada, Spain.

Nuhu, A.-R., M. Nabil, Y. Ayalew, V. Hemmati, A. Homaifar, and E. Tunstel (2023). Local (per-input) robustness based guided adversarial training of deep neural networks. In *2023 IEEE 14th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, pp. 0182–0191.

Pang, T., C. Du, Y. Dong, and J. Zhu (2018). Towards robust detection of adversarial examples. In *Advances in neural information processing systems*, Volume 31.

Pang, T., M. Lin, X. Yang, J. Zhu, and S. Yan (2022). Robustness and accuracy could be reconcilable by (proper) definition. In *International Conference on Machine Learning*, pp. 17258–17277. PMLR.

Papernot, N., P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami (2016). The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pp. 372–387. IEEE.

Papernot, N., P. McDaniel, X. Wu, S. Jha, and A. Swami (2016). Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE symposium on security and privacy (SP)*, pp. 582–597. IEEE.

Park, L. H., J. Kim, M. G. Oh, J. Park, and T. Kwon (2024). Adversarial feature alignment: Balancing robustness and accuracy in deep learning via adversarial training. In *arXiv preprint arXiv:2402.12187*.

Peng, S., W. Xu, C. Cornelius, M. Hull, K. Li, R. Duggal, M. Phute, J. Martin, and D. H. Chau (2023). Robust principles: Architectural design principles for adversarially robust cnns. In *arXiv preprint arXiv:2308.16258*.

Qian, H. and M. N. Wegman (2018). L2-nonexpansive neural networks. In *arXiv preprint arXiv:1802.07896*.

Qian, Y., S. Huang, B. Wang, X. Ling, X. Guan, Z. Gu, S. Zeng, W. Zhou, and H. Wang (2022). Robust network architecture search via feature distortion restraining. In *European Conference on Computer Vision*, pp. 122–138. Springer.

Qian, Z., K. Huang, Q.-F. Wang, and X.-Y. Zhang (2022). A survey of robust adversarial training in pattern recognition: Fundamental, theory, and methodologies. *Pattern Recognition 131*, 108889.

Qin, C., J. Martens, S. Gowal, D. Krishnan, K. Dvijotham, A. Fawzi, S. De, R. Stanforth, and P. Kohli (2019). Adversarial robustness through local linearization. In *Advances in Neural Information Processing Systems*, Volume 32.

Rade, R. and S.-M. Moosavi-Dezfooli (2021). Helper-based adversarial training: Reducing excessive margin to achieve a better accuracy vs. robustness trade-off. In *ICML 2021 Workshop on Adversarial Machine Learning*.

Rakin, A. S., J. Yi, B. Gong, and D. Fan (2018). Defend deep neural networks against adversarial examples via fixed and dynamic quantized activation functions. In *arXiv preprint arXiv:1807.06714*.

Rebuffi, S.-A., S. Gowal, D. A. Calian, F. Stimberg, O. Wiles, and T. Mann (2021). Fixing data augmentation to improve adversarial robustness. In *arXiv preprint arXiv:2103.01946*.

Ren, H., T. Huang, and H. Yan (2021). Adversarial examples: attacks and defenses in the physical world. *International Journal of Machine Learning and Cybernetics 12*(11), 3325–3336.

Ren, K., T. Zheng, Z. Qin, and X. Liu (2020). Adversarial attacks and defenses in deep learning. *Engineering 6*(3), 346–360.

Rice, L., E. Wong, and Z. Kolter (2020). Overfitting in adversarially robust deep learning. In *International conference on machine learning*, pp. 8093–8104. PMLR.

Russakovsky, O., J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision 115*, 211–252.

Samangouei, P., M. Kabkab, and R. Chellappa (2018). Defense-gan: Protecting classifiers against adversarial attacks using generative models. In *International Conference on Learning Representations*.

Sandler, M., A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520.

Sehwag, V., S. Wang, P. Mittal, and S. Jana (2020). Hydra: Pruning adversarially robust neural networks. *Advances in Neural Information Processing Systems 33*, 19655–19666.

Shafahi, A., M. Najibi, M. A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L. S. Davis, G. Taylor, and T. Goldstein (2019). Adversarial training for free! In *Advances in Neural Information Processing Systems*, Volume 32.

Shah, M., A. Kashaf, and B. Raj (2024). Training on foveated images improves robustness to adversarial attacks. In *Advances in Neural Information Processing Systems*, Volume 36.

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

Sooksatra, K., G. Hamerly, and P. Rivas (2023). Is ReLU adversarially robust? In *LatinX in AI Workshop at ICML 2023*.

Sooksatra, K. and P. Rivas (2021). Enhancing adversarial examples on deep q networks with previous information. In *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 01–07. IEEE.

Sooksatra, K. and P. Rivas (2022). Evaluation of adversarial attacks sensitivity of classifiers with occluded input data. *Neural Computing and Applications 34*(20), 17615–17632.

Sooksatra, K., P. Rivas, and J. Orduz (2021). Evaluating accuracy and adversarial robustness of quanvolutional neural networks. In *2021 International Conference on Computational Science and Computational Intelligence (CSCI)*, pp. 152–157. IEEE.

Suzuki, S., S. Yamaguchi, S. Takeda, S. Kanai, N. Makishima, A. Ando, and R. Masumura (2023). Adversarial finetuning with latent representation constraint to mitigate accuracy-robustness tradeoff. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4367–4378. IEEE.

Szegedy, C., V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826.

Szegedy, C., W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus (2013). Intriguing properties of neural networks. In *arXiv preprint arXiv:1312.6199*.

Tao, Y. (2023). Meta learning enabled adversarial defense. In *2023 IEEE International Conference on Sensors, Electronics and Computer Engineering (ICSECE)*, pp. 1326–1330. IEEE.

Tramer, F. and D. Boneh (2019). Adversarial training and robustness for multiple perturbations. In *Advances in neural information processing systems*, Volume 32.

Tramèr, F., A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel (2018). Ensemble adversarial training: Attacks and defenses. In *International Conference on Learning Representations*.

Vorácek, V. and M. Hein (2023). Improving l1-certified robustness via randomized smoothing by leveraging box constraints. In *International Conference on Machine Learning*, pp. 35198–35222. PMLR.

Vorobeychik, Y. (2023). The many faces of adversarial machine learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Volume 37, pp. 15402–15409.

Wang, D., W. Jin, and Y. Wu (2023). Between-class adversarial training for improving adversarial robustness of image classification. *Sensors 23*(6), 3252.

Wang, J., C. Wang, Q. Lin, C. Luo, C. Wu, and J. Li (2022). Adversarial attacks and defenses in deep learning for image recognition: A survey. In *Neurocomputing*. Elsevier.

Wang, R., Y. Li, and S. Liu (2023). Exploring diversified adversarial robustness in neural networks via robust mode connectivity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2345–2351.

Wang, Y., D. Zou, J. Yi, J. Bailey, X. Ma, and Q. Gu (2019). Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations*.

Wang, Z., T. Pang, C. Du, M. Lin, W. Liu, and S. Yan (2023). Better diffusion models further improve adversarial training. In *International Conference on Machine Learning*, pp. 36246–36263. PMLR.

Weitzner, D. and R. Giryes (2023). On the relationship between universal adversarial attacks and sparse representations. *IEEE Open Journal of Signal Processing 4*, 99–107.

Wong, E. and Z. Kolter (2018). Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International conference on machine learning*, pp. 5286–5295. PMLR.

Wong, E., L. Rice, and J. Z. Kolter (2020). Fast is better than free: Revisiting adversarial training. In *arXiv preprint arXiv:2001.03994*.

Wu, B., J. Chen, D. Cai, X. He, and Q. Gu (2021). Do wider neural networks really help adversarial robustness? *Advances in Neural Information Processing Systems 34*, 7054–7067.

Wu, Y., Y. Guo, D. Chen, T. Yu, H. Xiao, Y. Guo, and L. Bai (2024). Boosting adversarial robustness via feature refinement, suppression, and alignment. In *Complex & Intelligent Systems*, pp. 1–21. Springer.

Xiao, H., K. Rasul, and R. Vollgraf (2017a). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. In *arXiv preprint arXiv:1708.07747*.

Xiao, H., K. Rasul, and R. Vollgraf (2017b). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. In *CoRR*, Volume abs/1708.07747.

Xie, C., M. Tan, B. Gong, A. Yuille, and Q. V. Le (2020). Smooth adversarial training. In *arXiv preprint arXiv:2006.14536*.

Xie, C., Y. Wu, L. v. d. Maaten, A. L. Yuille, and K. He (2019). Feature denoising for improving adversarial robustness. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 501–509.

Xu, W., D. Evans, and Y. Qi (2017). Feature squeezing: Detecting adversarial examples in deep neural networks. In *arXiv preprint arXiv:1704.01155*.

Yang, Y., C. Lin, X. Ji, Q. Tian, Q. Li, H. Yang, Z. Wang, and C. Shen (2023). Towards deep learning models resistant to transfer-based adversarial attacks via data-centric robust learning. In *Association for the Advancement of Artificial Intelligence (AAAI)*.

Yang, Y., G. Zhang, D. Katabi, and Z. Xu (2019). Me-net: Towards effective adversarial robustness with matrix estimation. In *arXiv preprint arXiv:1905.11971*.

Yang, Z., Q. Xu, W. Hou, S. Bao, Y. He, X. Cao, and Q. Huang (2023). Revisiting auc-oriented adversarial training with loss-agnostic perturbations. *IEEE Transactions on Pattern Analysis and Machine Intelligence 45*(12), 15494–15511.

Yu, C., J. Chen, Y. Wang, Y. Xue, and H. Ma (2023). Improving adversarial robustness against universal patch attacks through feature norm suppressing. In *IEEE Transactions on Neural Networks and Learning Systems*. IEEE.

Zhang, B., T. Cai, Z. Lu, D. He, and L. Wang (2021). Towards certifying robustness using neural networks with l-dist neurons. In *International Conference on Machine Learning*.

Zhang, B., D. Jiang, D. He, and L. Wang (2022). Boosting the certified robustness of l-infinity distance nets. In *International Conference on Learning Representations*.

Zhang, H., Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. Jordan (2019). Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pp. 7472–7482. PMLR.

Zhang, J., J. Zhu, G. Niu, B. Han, M. Sugiyama, and M. Kankanhalli (2021). Geometry-aware instance-reweighted adversarial training. In *International Conference on Learning Representations*.

Zhang, Y., R. Cai, T. Chen, G. Zhang, H. Zhang, P.-Y. Chen, S. Chang, Z. Wang, and S. Liu (2023). Robust mixture-of-expert training for convolutional neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 90–101.

Zhao, Z., G. Chen, J. Wang, Y. Yang, F. Song, and J. Sun (2021). Attack as defense: Characterizing adversarial examples using robustness. In *Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis*, pp. 42–55.