Adversarial Robustness for Deep Learning

Matthew Whelan*
mw3shc@virginia.edu
University of Virginia
Charlottesville, Virginia, USA

Sidhardh Burre
ssb3vk@virginia.edu
University of Virginia
Charlottesville, Virginia, USA

ABSTRACT

Deep Neural Networks (DNNs), specifically Convolutional Neural Networks (CNNs) and Vision Transformers, have revolutionized various fields by improving the processing and analysis of high-dimensional data, such as images. A significant challenge in deploying these models is their vulnerability to adversarial examples—perturbations designed to cause models to make incorrect predictions. This paper investigates the adversarial robustness of different deep learning architectures, including CNNs and Vision Transformers, by employing various adversarial training methods such as Fast Gradient Sign Method (FGSM), Basic Iterative Method (BIM), and training with adversarial examples generated through Diffusion Probabilistic Models (DDPM). We evaluate these models on standard datasets like CIFAR-10, assessing their performance against both clean and adversarially perturbed data. Our findings demonstrate significant variances in how each model architecture responds to different adversarial training techniques. While CNNs generally show improvements in adversarial robustness at the cost of clean data accuracy, Vision Transformers exhibit a nuanced resilience, highlighting the importance of architectural considerations in designing robust AI systems. Additionally, our results underline the effectiveness of stochastic adversarial training and the potential of generative models like DDPM to enhance model robustness. This comprehensive analysis provides insights into optimizing deep learning models for increased security and reliability in real-world applications.

ACM Reference Format:
Matthew Whelan and Sidhardh Burre. 2024. Adversarial Robustness for Deep Learning. In *Proceedings of Software Analysis (CS6888 '24).* ACM, New York, NY, USA, 12 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Deep Neural Networks (DNNs) are a subclass of machine learning models that have gained immense popularity due to their ability to perform complex tasks with high accuracy. They differ for traditional neural networks with a structure composed of stacked layers of nodes. With layers of interconnected nodes or "neurons" that can learn to recognize patterns from large amounts of data, DNNs display the ability to improve continuously. This nature of Deep Learning models has made them an emerging technology in applications across various sectors like health, autonomous vehicles and security.

Due to DNN's capabilities, they are applied across fields, in the reconstruction of neural circuits [15], the analysis of DNA mutations [40], predicting the structure-activity of drug molecules[26], and analyzing particle accelerator data [4].

Despite the application of DNNs in a variety of fields, a core application of DNNs is in computer vision, or the discipline of processing, analyzing, and interpreting images via a computer. Computer vision is difficult due to the size of the input. Every image lies on an extremely high-dimensional manifold. An n by n image effectively exists in a $\mathbb{R}^{n*n}$ space. Modern consumer cellphones can create 4K images where the horizontal dimension is guaranteed to have a 4000 pixel width. Although this problem can be solved with larger models, this approach is infeasible. As the model size increases, it is more likely to encounter vanishing/exploding gradients [1][7]. Therefore, greater research was directed towards improving DNN performance on high-dimensional input.

A key innovation that improved DNN accuracy on images was convolutions. In a paper by Krizhevsky et al. [19], The authors hypothesize that convolutional layers "solve the localization and segmentation tasks" allowing later layers to only focus on representation. In their application of Deep Convolutional Neural Networks to an image recognition task, the authors demonstrate SoTA computer vision performance. Convolutional neural nets (CNN) have since been used in many industries for their localization and segmentation abilities [6] [24] [13] [31]. These capabilities have allowed neural networks to analyze large-scale images without significant increases in size. Subsequently, CNNs have become the basis for future computer vision tasks.

Further, the use of CNNs helped partially alleviate the vanishing/exploding gradients problem, stabilizing larger models. In a paper by Goodfellow et al., the authors demonstrate how the depth of the model directly correlates to model performance as seen in figure 1 without a dip in accuracy [8].

But as these models grew deeper, it became more difficult to obtain reasonable training accuracies as deep networks begin to converge, a degradation problem occurs. As the size of the network increases, the accuracy gets saturated and then degrades rapidly. But, this degradation does not occur due to over fitting and adding more layers to a sufficiently deep model leads to higher training error [35] [12]. To address these issues, the concept of Residual Neural Networks (RNNs) was formulated. Here, instead of layers directly feeding-forward information, there would be short-cut connections between layers enabling pass-through input as seen in Figure 2.

*Both authors contributed equally to this research.

CS6888 '24, Jan 17–May 04, 2024, Charlottesville, VA

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

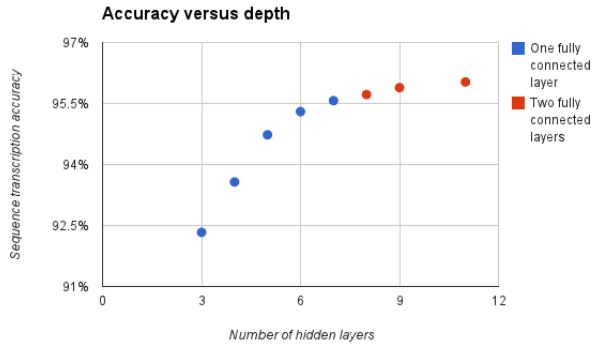

Figure 1: Caption

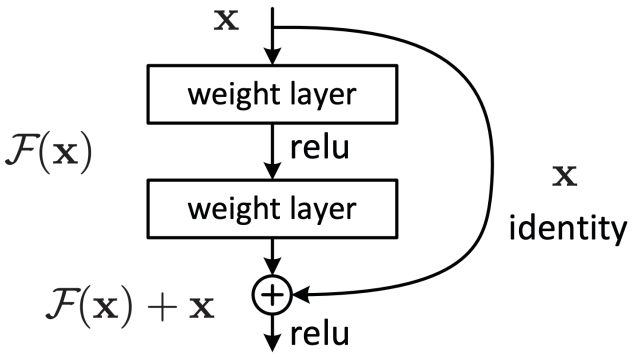

Figure 2: Caption

In a paper by He et al., the authors demonstrate the power of residual connections. Despite the addition of further layers, the models experience the expected improvement in performance as opposed to the decrease in performance seen in traditional networks [14]. The conclusion of the He et al. paper can be seen clearly in Figure 3. Within this figure, the left graph shows how plain networks display a trend of increasing error rate as the number of model layers increases. But, the middle graphs displays a trend of residual networks decreasing the error rate as the number of model layers increases. This work by He et al. demonstrated the ability for residual connections to preserve model performance at scale.

Clearly, the field of computer vision has progressed far, addressing the problems of vanishing/gradient descent and developing scalable architectures, setting records for accuracies across all challenges. Despite their impressive capabilities, DNNs face significant challenges in the form of adversarial examples, Szegedy et al. [36] first discovered this problem. The authors demonstrated that despite high accuracies, deep networks are highly susceptible to adversarial attacks that only slightly perturb images. These attacks cause models to report high confidence on an incorrect prediction. Further, this class of attacks demonstrate transferrability. An attack that works on one model is likely to work on other models as well. This vulnerability is particularly concerning in applications involving critical safety and security decisions, such as in autonomous driving and security surveillance systems, where incorrect interpretations can have dire consequences.

2 BACKGROUND

In this section, we deep-dive into some of the technical aspects of adversarial attacks. Therefore, some of the common technical terms will be defined.

- *Adversarial example/image* is a modified image that undergoes some perturbation process such that a target machine learning model cannot properly label the image.
- *Adversarial perturbation* is the noise or operation applied to the original image to produce an adversarial example/image.
- *Adversarial training* is a type of neural network training that trains the network on adversarial examples instead of clean/standard examples.
- *Black-box attacks* feed a target model with adversarial examples without knowledge of the target model. In most instances, it is understood that the black-box attacker has little to no information about the target model.
- *Quasi-imperceptible* perturbations modify images in a manner that is imperceptible to humans.
- *White-box attacks* have complete knowledge about the model including parameter values, architecture, training method, and in some contexts training data as well.

Because the focus of this paper is how different adversarial training methods can improve the adversarial robustness of a variety of models as seen in Goodfellow et al. [9], it is essential to have an understanding of adversarial training. One such method of adversarial training is training the model directly on adversarially perturbed images. In this study, the Fast Gradient Signed Method (FGSM) and Basic Iterative Methods (BIM) are utilized to generate adversarially perturbed images. An alternative method of improving adversarial robustness is by training on non-conditainally diffusion-model generated images as shown by Rebuffi et al. [30]. Subsequent papers have demonstrated further improvements in adversarial robustness as a result of training on diffusion-models [39].

2.1 Fast Gradient Sign Method

A classical adversarial example is shown in Figure 4, where an image of a Panda is distorted via adding an imperceptibly small vector and the classification changes dramatically. Specifically, this is done via the "fast gradient" method, first introduced in [9]. This method efficiently generates an adversarial perturbation for a given image X:

$$X_{\text{adv}} = X + \epsilon \cdot \text{sign}(\nabla_X J(X, y_{\text{true}})) \tag{1}$$

where $\nabla_X J$ computes the gradient of the cost function around the current value of the model parameters, sign(.) denotes the sign function, and ϵ is a small scalar value that restricts the norm or "distance" of the perturbation.

FGSM is motivated by exploiting the linearity of DNNs in their high dimensional space. According to the linearity hypothesis, posited by Goodfellow et al. [9], DNNs are encouraged to create linear sub-partitions of their high dimensional space for computational gains that in turn leave them susceptible to simple perturbations.

A subsequent paper by Kurakin et al. [20] demonstrated that on the ImageNet dataset, the error rate on candidate adversarial

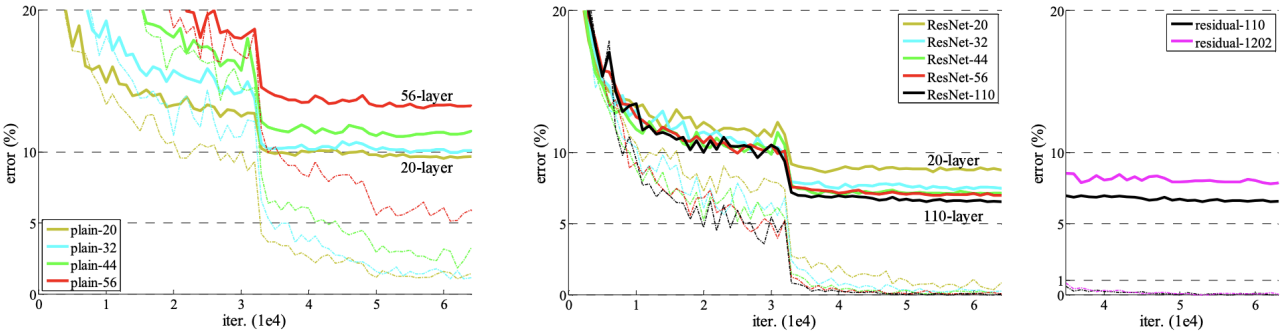

Figure 3: Training on CIFAR-10. Dashed lines denote training error, and bold lines denote testing error. Left: plain networks. The error of plain-110 is higher than 60% and not displayed. Middle: ResNets. Right: ResNets with 110 and 1202 layers.

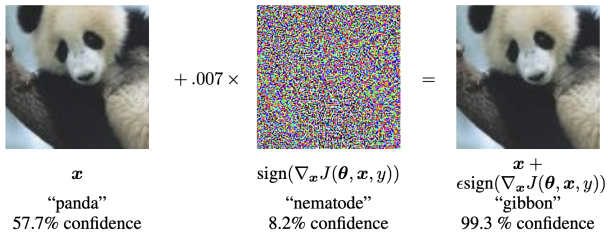

Figure 4: Adversarial Example from [9]

images for FGSM generated examples is betweeen 63% and 69% for $\epsilon \in [2, 32]$. The authors also demonstrated a method to deterministically ensure the mis-prediction class of the generated adversarial example. Instead of using the y_{true} label of the image in equation 1, they used the label of the least likely class for the network. The computed perturbation can then be applied to the original image to generate a targeted adversarial example.

2.2 Basic Iterative Method

FGSM perturbs images by taking a single large step in the direction that increases classification loss. The next intuitive extension is to proceed iteratively, taking multiple small steps with a re-adjusted direction after every step. The Basic Iterative Method iteratively computes the step direction:

$$X_{\text{adv}}^{(N+1)} = \text{Clip}_{X,\epsilon}\left(X_{\text{adv}}^{(N)} + \alpha \cdot \text{sign}(\nabla_X J(X_{\text{adv}}^{(N)}, y_{\text{true}}))\right) \quad (2)$$

where $X_{\text{adv}}^{(N+1)}$ denotes the perturbed image a the $N+1^{th}$ iteration, $Clip_{X,\epsilon}$ clips the image in its argument at ϵ and α determines the step size.

This method was introduced by Kurakin et al. [21]. In our study, we used $\alpha = 1$, i.e., we changed the value of each pixel only by 1 on each step. The number of iterations was selected via $\lfloor \min(\epsilon + 4, 1.25\epsilon) \rfloor$. This number of iterations was chosen heuristically; it is sufficient for the adversarial example to reach the edge of the ϵ max-norm ball but restricted enough to keep the computational cost of experiments manageable. In later contexts, we refer to this method as the "basic iterative" method.

3 RELATED WORK

3.1 Adversarial Training

In a paper by Madry et al. [27], the authors demonstrate how adversarial examples can be used to improve the adversarial accuracy of models. Namely, they show how incorporating adversarially perturbed examples into a model's training data can improve that model's adversarial robustness. Further the authors present the following conclusions in regards to model capacity and attack methods:

(1) FGSM adversaries don't increase robustness (for large ϵ)
(2) Weak models may fail to learn non-trivial classifiers
(3) More capacity and stronger adversaries decrease transferability

The first conclusion is a symptom of FGSM. Kurakin et al. [20] explain that FGSM creates very restricted examples that depart from the data manifold. Thus, when the model is trained on these examples, it is forced to classify images that are not part of the original data distribution, resulting in model over fitting and losing accuracy on the original data distribution.

The second conclusion indicates that when smaller, weaker models are trained on a strong adversary such as PGD/BIM, the model fails to learn anything. Although the model can converge to reasonable accuracy through standard training, when trained adversarially, it converges to fixed class prediction. The authors hypothesize that this is a result of sacrificing standard performance for adversarial robustness.

The final conclusion originates from the fact that higher capacity models can express more complex partition functions on the input domain. As such, the authors demonstrate that adversarial examples that are used to trick one complex model cannot be used to trick another. This indicates a lack of transferability of adversarial examples from complex models to other complex models. But, examples that work on complex models can be reasonably expected to work on simpler, less complex models.

Subsequent research has focused on balancing adversarial robustness with standard accuracy. This is most exemplified by the work by Zhang et al. [42] who propose TRADES to balance adversarial and standard accuracy against l_∞ norm-bounded perturbations.

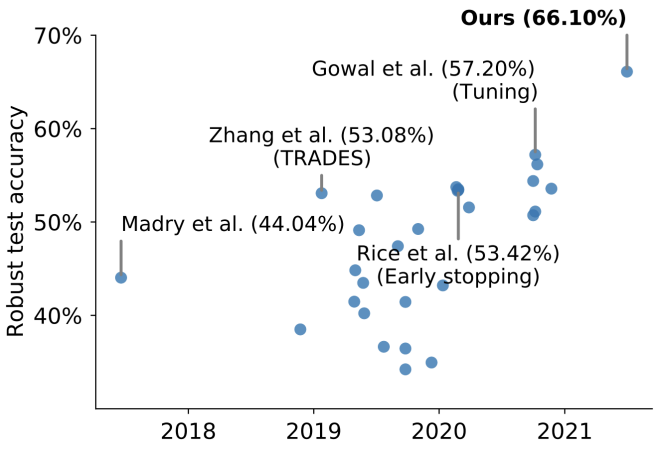

Figure 5: Robust accuracy of models against AUTOATTACK on CIFAR-10 from Gowal et al. [10]

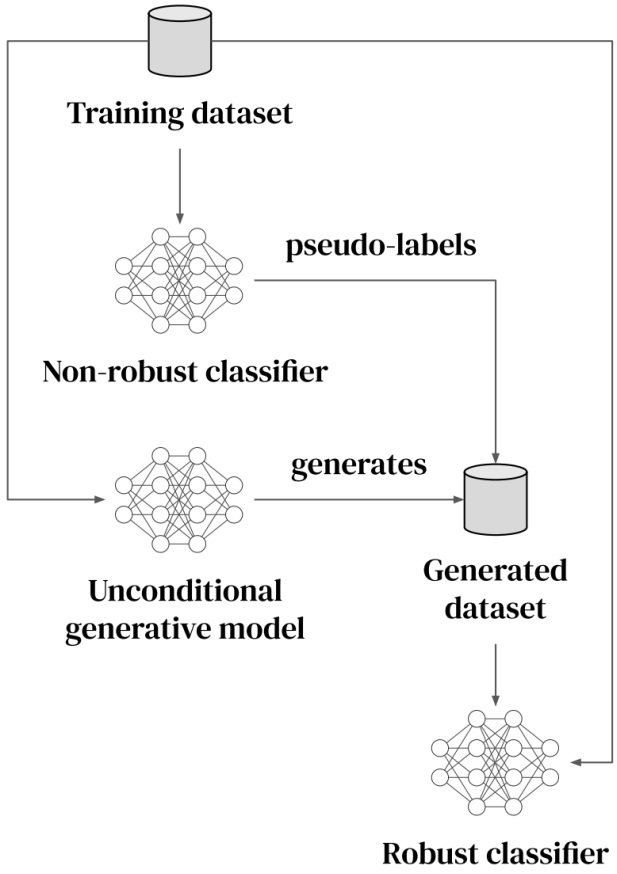

Figure 6: Approach as outlined by Gowal et al. [10]

This work was supported by Rice et al. [32] who demonstrate an efficient improvement to TRADES via early stopping.

3.2 Improving Adversarial Robustness

The proposal of using adversarial training to improve robustness by Madry et al. [27] is widely regarded as one of the most successful ways to train robust DNNs and has been augmented in a number of ways. Papers by Carmon et al. [3], Najafi et al.[29], Uesato et al.[37], Zhai et al. [41], simultaneously proposed the use of additional unlabeled external data within this adversarial training progress. Yet, without additional data, it became difficult to boost robust accuracy significantly. Yet, in a paper by Gowal et al. [10], the authors demonstrate SoTA adversarial robustness using generative models to generate data as seen in Figure 5.

Further, they propose a training pipeline that enables adversarial training on unlabeled data as seen in Figure 6. This approach trains a generative model and a non-robust classifier on the initial dataset. The non-robust classifier is used to provide pseudo-labels to the images produced by the generative model. Finally, the generated and original data are used as training data to train an adversarially robust model.

3.3 Generative Models

A paper by Sohl-Dickstein et al. [34] presents a novel way to define proabilistic models to allow for greater flexibility. Their method utilizes a Markov chain to incrementally convert one distribution to another, an idea borrowed from non-equilibrium statistical physics by Jarzynski [17]. The authors build a generative Markov chain that converts a simple known distribution into a target distribution using a diffusion process. Their key innovation comes from the following: instead of using the Markov chain to evaluate a model that has been defined, they instead define the probabilistic model as the endpoint of the Markov chain. Because each step in the chain has an analytically evaluable probability, the entire chain can be analytically evaluated as well. This key improvement allowed for the inception of EDM [18] and Deep Diffusion Probabilistic Models (DDPM) [16]

In a paper published by Ho et al. [16], the authors present significant progress in diffusion probabilistic models originally presented by Sohl-Dickstein et al. [34]. As mentioned before, a diffusion model is a parameterized form of a Markov chain trained on variational inference to produce image samples that match the original dataset. Transitions along the Markov chain are learned to reverse the diffusion process; creating a Markov chain that incrementally peppers an image with Gaussian noise until the image is destroyed. The authors find that when the diffusion consists of small amounts of Gaussian noise, it is sufficient to set the sampling chain transitions to conditional Gaussians as well, thereby enabling a simple neural network parameterization. This parameterization is Ho et al.'s key contribution that enables the DDPM model to generate SoTA image generation quality.

3.4 Vision Transformers

The Transformer was introduced in the landmark paper by Vaswani et al. [38], and it quickly became a popular choice for natural language processing tasks like machine translation. This architecture diverged from previous models by using self-attention mechanisms, which weigh the significance of different words, irrespective of

their position in the input sequences. This allows Transformers to excel in tasks that require understanding the context of words in long sequences, making them highly efficient and scalable.

Building on the success of Transformers in NLP, Dosovitskiy et al. [5] created the Vision Transformer (ViT), which adapts the Transformer architecture for image processing tasks. In traditional computer vision, Convolutional Neural Networks (CNNs) have been predominant due to their ability to hierarchically extract spatial features from images. Instead of processing pixels directly, ViT divides an image into fixed-size patches, treats these patches as tokens (similar to words in NLP), and processes these sequences of patches using a standard Transformer model. The paper shows that this approach can rival the performance of state-of-the-art CNNs on various image recognition benchmarks, like ImageNet and CIFAR-100, while being more computationally efficient.

The Swin Transformer from Liu et al. [22] builds upon the principles established by the Vision Transformer (ViT); instead of treating images through a non-hierarchical approach, using fixed-size patches as input tokens, the Swin Transformer introduces a hierarchical architecture that uses a "sliding window" mechanism for computing self-attention. This allows Swin Transformers to handle varying scales of visual data more effectively than previous Vision Transformer approaches. This is the first work that demonstrated Transformers can be used as a generic vision backbone, outperforming previous methods in a range of computer vision tasks beyond just image classification.

Based on the success of Vision Transformers, Liu et al. [23] attempted to understand the differences between traditional CNNs and ViTs by starting with a standard ResNet model and "modernizing" the architecture to the construction of a hierarchical vision transformer (like Swin). This led to the creation of the ConvNeXt, which adjusted the CNN architecture without the use of self attention mechanisms: like changing the stage compute ratio, adopting a patchify stem, and employing larger kernel sizes in the convolutional layers. These design decisions as well as specific training techniques taken from the Swin Transformer paper allow ConvNeXt to surpass the performance of more complex Transformer-based models with better computational efficiency.

There is also notable modern work that investigate Vision Transformers specifically as it relates to adversarial robustness. Shao et al. [33] examined the inherent architectural features of ViTs, such as self-attention mechanisms, which may contribute to their resilience against adversarial attacks. As such, they provides empirical evidence suggesting that ViTs may offer an improved baseline robustness over CNNs, potentially due to their different processing and integration of features across an image . Mahmood et al. [28] investigated the susceptibility of Vision Transformers to various adversarial attacks and contrasts their performance with that of CNNs for both white-box and black-box attacks. They further look at the transferability of adversarial examples between CNNs and ViTs, finding that adversarial examples do not readily transfer, which facilitates the exploration of ensemble models combining both architectures. The findings indicate that such ensembles can significantly enhance robustness in black-box scenarios but not in white-box scenarios.

Even more recently (publications released earlier this year), there has been work in Vision Transformer based Diffusion models. Specifically, models have been designed for unrestricted resolution and aspect ratio generation [25] and for generally higher fidelity images with significantly better parameter efficiency [11].

4 APPROACH

The purpose of this project is to analyze adversarial robustness across a variety of model architectures and adversarial training methods. To that end, we selected the image vision models detailed in Table 1. Within our selection, we have a Deep Convolutional Network in VGG11_BN, two residual networks in ResNet18 and Wide_ResNet50_2, two vision transformers in ViT_B_16 and Swin_Tiny, and a modern image model ConvNeXt_Tiny. Due to computational constraints, we were unable to use any other or larger models.

Model	Top 1	Top 5	Params	GFLOPS	Date
VGG11_BN	70.37	89.81	132.9M	7.61	Apr 2015
ResNet18	69.758	89.078	11.7M	1.81	Dec 2015
Wide_ResNet50_2	81.602	95.758	68.9M	11.4	Jun 2017
ViT_B_16	81.072	95.318	86.6M	17.56	Oct 2020
Swin_Tiny	81.474	95.776	28.3M	4.49	Aug 2021
ConvNeXt_Tiny	82.52	96.146	28.6M	4.46	Jan 2022

Table 1: Pretrained model performance metrics including publication (SoTA) dates

Each of the models found in Table 1 had available pretrained weights that were downloaded via `torchvision.models` package in Python. The pretrained weights come from training on the ImageNet dataset, where images are resized to 256, center cropped to 224 and then scaled and normalized. For consistency, we decided to use the CIFAR-10 dataset, which is 32x32 images, and did the same preprocessing operations as the ImageNet pretrained procedure. Note also that the number of output classes for ImageNet is 1000, while CIFAR-10 has only 10, so the output layer of each of the pretrained models was adjusted accordingly. Each of the models then underwent 5 epochs of fine-tuning on the CIFAR-10 dataset. This number of epochs ensured that the models were not overfit to the training data.

For adversarial training methods, we used FGSM, BIM, and DDPM-based training. For FGSM based adversarial training, the complete training dataset was perturbed with $\epsilon = 0.1$ and the models were trained on this perturbed data for 10 epochs. Throughout training, the models were evaluated on an FGSM perturbed version of the test dataset.

For BIM based training, an $\alpha = 0.01$ was used with 10 iterations as well as an $\epsilon = 0.03$. The combination of an $\alpha = 0.01$ and 10 iterations ensures that the path-wise change in BIM is comparable to the net change in FGSM ($\epsilon = 0.1$). This combination attempts to create an apples-to-apples comparison between the FGSM and BIM methods. The $\epsilon = 0.03$ limits the level of perturbation at each step, creating subtler perturbations that should remain close to the original training data's manifold. After the training data set is perturbed, the respective model is trained on the perturbed dataset for 10 epochs.

For DDPM training, Google's open-source DDPM pretrained on CIFAR-10 via Hugging Face was used. Because this is an unconditional modle, it creates images without any specificed conditions or inputs that define what the image should display. Therefore, a non-robust classifier was used to classify the DDPM images and the robust classifier was trained on these new pseudo-labels. We generate pseudo-labels using the same model adversarially trained on FGSM. DDPM images are generated over 100 iterations in batches at training time and used as training data. The optimal number of DDPM iterations was found through trying multiple values of iterations and evaluating the transfer accuracy of the pseudo-labels.

One of the first experiments was aimed at evaluating the impact of the `num_inference_steps` in the DDPM Pipeline, where more denoising steps usually lead to a higher quality image at the expense of slower inference. This was done using a basic pre-trained ResNet18 model for 1000 epochs of DDPM generated data with inference steps $\in [1, 10, 50, 100, 500]$. Then, the model was evaluated on 100 epochs of data in the same inference step range to determine performance.

An additional training approach called "Stochastic Adversarial Training" was used in this experiment. In other experiments, it was noted that clean accuracy has a tendency to decrease when models are hardened only against adversarial attacks. To address this limitation, stochastic adversarial training was devised to train models in a more balanced method. For each epoch of model training, the model is either exposed to BIM perturbed data or clean data with ϵ probability (where ϵ is the probability that the model is exposed to clean data). Note that the model is still fine-tuned on CIFAR10 before Stochastic Adversarial Training.

For following combination methods are attempted: FGSM + BIM and FGSM + DDPM. BIM + DDPM was not attempted due to the computational cost of BIM.

5 RESULTS

5.1 DDPM Optimization

Although DDPM is highly efficient, it still requires some optimization in terms of the number of iterations. At the start of this experiment, we were unsure as to the number of iterations that would be suitable for our experiments. As the number of iterations had a direct impact on the validity of our results, initial experiments were directed at optimizing the number of iterations of DDPM to use for our training.

To produce Figure 7, a pretrained ResNet 18 model was used as both the pseudo labeller and the test model. As seen in Figure 7, there are diminishing returns after 50-100 steps of DDPM inference, the created images do not drastically change DDPM transfer accuracy.

5.2 FGSM Adversarial Training

The first set of experiments tested adversarial robustness after FGSM attacks. The training data was perturbed for each model with FGSM method and the models were trained on their respective perturbed data. The performance from this method is shown below in Table 2

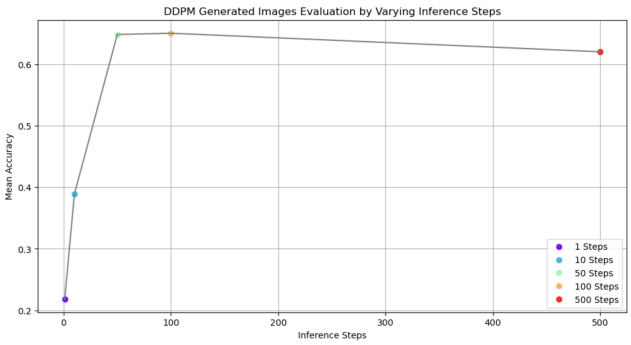

Figure 7: DDPM Varying Inference Steps

Model	Pre-Adv		FGSM	
	Clean	Adv	Clean	Adv
VGG11_bn	93%	21%	87%	74%
ResNet 18	94%	25%	91%	73%
Wide ResNet 50	96%	35%	93%	80%
ViT	97%	36%	60%	91%
Swin_t	97%	23%	97%	69%
ConvNext	97%	25%	84%	92%

Table 2: Test Accuracy before and after FGSM adversarial training

This experiment shows a general decrease in clean accuracy post-adversarial training, but a larger increase in adversarial accuracy. This increase is particularly notable in CNN based models such as VGG11_bn, ResNet18, and Wide ResNet 50, which see adversarial accuracy increases of 53%, 48%, and 45%, respectively. The transformer-based models (ConvNext and ViT but not Swin_t) have more dramatic fluctuations in both clean and adversarial accuracies, suggesting that the impact of FGSM training varies significantly across different architectures. The model deltas are summarized in Table 3

Model	FGSM	
	Clean Acc Δ	Adv Acc Δ
VGG11_bn	-6%	+53%
ResNet 18	-3%	+48%
Wide ResNet 50	-3%	+45%
ViT	-38%	+54%
Swin_t	0%	+46%
ConvNext	-13%	+67%

Table 3: Test Accuracy difference due to FGSM adversarial training

A clear outlier in Table 3 is the ViT model. It displays a drastic decrease in clean accuracy with only a middling improvement in

adversarial accuracy. Another outlier is the ConvNext model the decrease in clean accuracy is the second largest (in magnitude) and is combined with the largest increase in adversarial accuracy. This may potentially be due to the power of this model and over fitting on the adversarial data. Interestingly, the Swin_t model displays no decrease in clean accuracy, displaying only an increase in adversarial accuracy. Although its final adversarial accuracy is lower than that of ViT and ConvNext's adversarial accuracies, its lack of decrease in clean accuracy may be indicative of certain architectural choices within the model.

5.3 BIM Adversarial Training

This experimental setup is similar to the previous one, except it is for the BIM method. It still involves taking the pre-trained models on CIFAR-10 and evaluating them on clean test set and BIM perturbed test data (represented as Adv in Table 4) before and after training 10 epochs on the perturbed train set.

Model	Pre-Adv		BIM Adv	
	Clean	Adv	Clean	Adv
VGG11_bn	93%	0%	63%	22%
ResNet 18	94%	0%	68%	31%
Wide ResNet 50	96%	0%	68%	28%
ViT_B_16	97%	0%	64%	7%
Swin_t	97%	0%	82%	9%
ConvNeXt	97%	0%	90%	11%

Table 4: Test Accuracy before and after BIM Adversarial Training

This experiment shows an overall decrease in clean accuracy post adversarial training with a low increase in adversarial accuracy. The CNN based models such as VGG, ResNet, and Wide ResNet show about a 30% adversarial accuracy whereas the transformer based models display a 10% adversarial accuracy. Again, the ViT model is a clear outlier among the vision transformer based models. Whereas Swin_t and ConvNext maintain a 80-90% clean accuracy, ViT dips far lower to 64% accuracy. The model deltas are summarized in Table 5.

Model	BIM	
	Clean Acc Δ	Adv Acc Δ
VGG11_bn	-30%	+22%
ResNet 18	-26%	+31%
Wide ResNet 50	-28%	+28%
ViT	-34%	+7%
Swin_t	-15%	+9%
ConvNext	-7%	+11%

Table 5: Test Accuracy difference due to BIM adversarial training

In terms of the magnitudinal difference in clean accuracy before and after training shown in Table 5, the ViT model is a clear outlier. Through BIM adversarial training, the ViT model sees the greatest difference in clean accuracy, more so than the other transformer models. Further, this decrease in adversarial training is more on par with the CNN based models. When inspecting the difference in adversarial accuracy, the Residual network based models see a greater increase in adversarial accuracy than a simple CNN model such as VGG11_bn. On the transformer side, these models see only about a third of an improvement in adversarial accuracy compared to their CNN counter parts.

5.4 FGSM + BIM Adversarial Training

To perform FGSM + BIM adversarial training, models that were adversarially trained on FGSM perturbed images were then subject to training on BIM perturbed images. The results are displayed in Table 6.

Model	Pre-Adv		FGSM		FGSM + BIM	
	Clean	Adv	Clean	Adv	Clean	Adv
VGG11_bn	93%	21%	87%	74%	52%	24%
ResNet 18	94%	25%	91%	73%	68%	37%
Wide ResNet 50	96%	35%	93%	80%	64%	39%
ViT	97%	36%	60%	91%	56%	5%
Swin_t	97%	23%	97%	69%	92%	15%
ConvNext	97%	25%	84%	92%	90%	11%

Table 6: Test Accuracy before and after BIM adversarial training

This experiment displays an overall and dramatic decrease in clean accuracy after BIM training. Further adversarial accuracy on the BIM perturbed data suffers greatly particularly on the more advanced transformer based models. Interestingly ConvNext and Swin_t show little to no drop in clean accuracy with ConvNext *increasing* in clean accuracy after BIM adversarial training. The model deltas are summarized in Table 7

Model	FGSM		FGSM + BIM	
	Clean Δ	Adv Δ	Clean Δ	Adv Δ
VGG11_bn	-6%	+53%	-41%	+3%
ResNet 18	-3%	+48%	-26%	+12%
Wide ResNet 50	-3%	+45%	-32%	+4%
ViT	-38%	+54%	-41%	-31%
Swin_t	0%	+46%	-5%	-8%
ConvNext	-13%	+67%	-7%	-14%

Table 7: Test Accuracy change due to FGSM and FGSM + BIM adversarial training compared to original values

Table 7 shows the difference in adversarial and clean accuracy between the indicated method and the model's initial accuracies. There is an apparent trend here where the adversarial accuracy increases for the CNN based models but in the transformer based models, adversarial accuracy increases. Further although most of the models see a decrease in clean accuracy, the Swin_t and ConvNext model see an order of magnitude smaller decrease in clean accuracy.

This suggests a compounded effect of continued adversarial training reducing general performance on clean data, emphasizing the trade-off between robustness to adversarial examples and accuracy on clean data.Here we note that it might be beneficial to limit the number of iterations or the magnitude of changes in BIM, ensuring that perturbations keep the data within realistic bounds or closer to the data manifold. Alternatively, incorporating regularization techniques that encourage the model to ignore high-frequency components (which are less likely to represent meaningful image content) could help maintain performance.

5.5 Stochastic Adversarial Training

Given the previous analysis on the effects of adversarial training methods FGSM and BIM, which showed that clean accuracy tends to decrease while models are specifically hardened against adversarial attacks, the proposed **Stochastic Adversarial Training** method aims to address these shortcomings by introducing a balanced approach to training on both clean and adversarial data specific to the BIM method. This balance is achieved by stochastically deciding whether a given batch of data will be processed as clean or adversarially perturbed based on a predefined probability threshold ϵ.

For this experiment an $\epsilon = 0.15$ was fine-tuned to best prioritize adversarial training and ensure that clean training will still happen 15% of the time.

First, to understand what sequence to run this experiment on, two methods were evaluated: the first training on FGSM data then training on BIM stochastically, and the second training on FGSM after stochastic training.

Method	Clean Δ	FGSM Δ	BIM Δ
FGSM Pre-Adv	-32%	+29.09%	+15%
No Pre-Adv (BIM, then FGSM)	-7%	+52%	+24%

Table 8: Comparison of ResNet Model Training Approaches

In evaluating these two approaches, it seems most beneficial to do the second method of BIM then FGSM in terms of maintaining clean accuracy and adversarial robustness. Training on Stohastic BIM and then FGSM significantly hinders the clean accuracy as well as the gained BIM Adversarial Accuracy. At this point, we decided to alter the experiment for clarity and hopefully improved robustness gain by taking out the final FGSM training and only focusing on Stochastic BIM training. In this way, the findings can be more directly compared with the results from Section 5.3, where only BIM Adversarial Training is applied to the models. Thus, this stochastic BIM method was tested on all of the models and the results are shown below:

Model	Pre		BIM	
	Clean	Adv	Clean	Adv
VGG11_bn	93%	0%	67%	30%
ResNet 18	94%	0%	53%	27%
Wide ResNet 50	96%	0%	54%	29%
ViT	97%	0%	59 %	8 %
Swin_t	97%	0%	93%	18%
ConvNext	97%	0%	89%	12%

Table 9: Performance before and after Stochastic BIM adversarial training

These results are similar to the normal BIM Adversarial Training experiments in Section 5.3. For clarity, the equivalent delta table is shown in Table 10 below, capturing the same information from Table 9.

Model	Clean Acc Δ	Adv Acc Δ
VGG11_bn	-26%	+30%
ResNet 18	-41%	+27%
Wide ResNet 50	-42%	+29%
ViT	-38%	+8%
Swin_t	-4%	+18%
ConvNext	-8%	+12%

Table 10: Change in Test Accuracy due to Stochastic BIM adversarial training

These outcomes underscore that while Stochastic BIM can significantly bolster adversarial robustness, the extent of clean accuracy retention varies widely between models. Modern models like Swin_t and ConvNext highlight the potential of Stochastic BIM to achieve a dual objective of maintaining high clean accuracy while substantially improving resistance to adversarial attacks. Conversely, traditional CNN models such as ResNet 18 and Wide ResNet 50 indicate a need for further optimization to balance robustness gains with acceptable levels of clean accuracy reduction.

5.6 DDPM Adversarial Training

This final experiment tested the adversarial robustness of a variety of models after being trained on unconditional DDPM generated CIFAR10 data that is labeled with pseudo labels from a fine-tuned model. For evaluation purposes, we tested FGSM pretrained models against FGSM perturbed data before and after the models were exposed to DDPM generated images. The results are displayed in Table 11:

Trend wise, the CNN based models see a drastic decrease in both clean and adversarial accuracy. In fact, they regress to being no better than random guessing on average. The transformer models on the other hand display steady increases in clean and

Model	Initial Clean	Initial FGSM	DDPM	After Clean	After FGSM
VGG	87%	74%	13%	12%	9%
WRN	93%	80%	5%	10%	10%
RN	91%	73%	16%	9%	10%
ViT	59%	90%	89%	56%	16%
Swin	90%	97%	81%	84%	25%
ConvNext	84%	92%	100%	90%	41%

Table 11: Performance of Models Before and After DDPM Training

adversarial accuracy across each of the models. Interestingly, the ConvNeXt model displays both an *increase* in clean accuracy and the smallest decrease in adversarial accuracy. The changes in clean and adversarial accuracy are shown in Table 12:

Model	DDPM	
	Clean Acc Δ	Adv Acc Δ
VGG11_bn	-75%	-65%
ResNet 18	-83%	-70%
Wide ResNet 50	-82%	-63%
ViT	-3%	-74%
Swin_t	-6%	-72%
ConvNext	+6%	-51%

Table 12: Test Accuracy difference due to DDPM adversarial training

Table 12 most clearly delineates how much of an outlier ConvNext is in terms of both the change in Clean Accuracy and FGSM Adversarial Accuracy. Whereas the other models see a decrease in clean accuracy and a 70% decrease in adversarial accuracy, ConvNext sees an *increase* in clean accuracy and the smallest decrease in adversarial accuracy of 51%. Trend-wise, the CNN based models post large decreases in clear accuracy of 80% whereas the transformer based models display a significantly smaller change in clean accuracy but both classes see an overall large dip in adversarial accuracy.

6 DISCUSSION

6.1 FGSM Adversarial Training

The FGSM adversarial training produced very different effects depending on the architecture of the underlying model. The shifting window transformer (Swin_t) model sees no decrease in clean accuracy while the ViT and ConvNext models see a decrease in adversarial accuracy as seen in Table 2. But, this lack of decrease in clean accuracy is accompanied by a corresponding lack of increase in adversarial accuracy. While the Swin_t model only obtains a final adversarial accuracy of 68%, the ViT and ConvNext model obtain accuracies of 90% and 92% respectively.

The reason why the Swin_t model does not experience a decrease in clean accuracy may be attributable to the high resolution feature maps that are used throughout the model. The shifting windows ensure that high-level features are mapped and analyzed instead of low-level features that ViT focuses on through its fixed windows [22] [5]. This also explains why Swin_t is unable to produce a on-par improvement in adversarial accuracy. Because the Swin_t model's architecture is optimized for classifying non-adversarial images, it is less generalizable and more difficult to fine tune with significant accuracy on adversarial datasets.

6.2 BIM Adversarial Training

This attack was much more complex than the FGSM method producing significantly more adversarial images, decreasing the ability for models to learn the properties of said images. This is evident by comparing Tables 7 and 5. In these tables, it is apparent that training on BIM perturbed data produces a far lower increase in adversarial accuracy and a greater decrease in clean accuracy. But initially, we hypothesized the opposite, that the FGSM perturbed data would result in perturbations that shift the data off the manifold, creating more adversarial images than BIM. But our findings indicate the opposite, models are able to adapt to FGSM's perturbations reasonably well. Instead, it is BIM that creates more adversarial images that are in fact more difficult for the models to learn from.

6.3 FGSM + BIM Adversarial Training

The experiment involving FGSM + BIM adversarial training demonstrates significant variations in the performance of different models under adversarial conditions. Initially, models trained with FGSM perturbations showed reasonable robustness, but subsequent BIM perturbations led to a notable decrease in clean data accuracy across most models. For instance, while CNN-based models such as VGG11_bn and ResNet 18 exhibited substantial drops in clean accuracy, transformer models like ViT showed even steeper declines, particularly in adversarial accuracy on BIM perturbed data. Notably, the ConvNext and Swin_t models maintained or even improved their clean accuracy, suggesting model-specific responses to the type of adversarial training applied.

The data summarized in the provided tables highlights these trends clearly. For CNN models, there is a general increase in adversarial accuracy post-FGSM training, which slightly extends after BIM training. However, transformer models do not follow this trend and generally show an increase in adversarial vulnerability after BIM training. The overall decrease in clean accuracy indicates a trade-off between enhancing adversarial robustness and maintaining performance on non-perturbed data. This suggests the potential necessity to moderate the intensity or frequency of BIM perturbations or to integrate regularization strategies aimed at preserving essential image features while ignoring adversarial noise, thus maintaining a balance between robustness and accuracy.

6.4 Stochastic Adversarial Training

Stochastic BIM training tends to yield better or comparable results in adversarial accuracy across most models compared to the regular BIM training in Section 5.3.

Traditional BIM adversarial training typically results in a significant drop in clean accuracy. This decline occurs because the model becomes overly specialized to resist adversarial perturbations, potentially at the cost of generalizing poorly on unperturbed data. However, with Stochastic BIM, where a model is exposed to

clean data 15% of the time, the results suggest a less pronounced decline in clean accuracy for certain models. Specifically, VGG11_bn shows a smaller decrease in clean accuracy with Stochastic BIM (-26%) compared to BIM (-30%), and for Swin_t there was only a -4% decrease compared to -15% with BIM.

ResNet 18 shows better adversarial robustness with BIM (+31%) compared to Stochastic BIM (+27%). Interestingly, while some models like Swin_t and ConvNext showed less improvement in adversarial robustness with Stochastic BIM compared to traditional BIM, the overall drop in clean accuracy was also less. This suggests a trade-off between maintaining higher clean accuracy and achieving maximum possible robustness against attacks.

Stochastic BIM presents a balanced approach by allowing models to not only learn from adversarially perturbed data but also from clean, unperturbed data. This balance helps in reducing the trade-off typically seen in adversarial training between clean accuracy and adversarial robustness. Models trained with Stochastic BIM tend to maintain better clean accuracy, making them more practical for real-world applications where both normal and perturbed data are encountered. However, the slight reduction in adversarial robustness with Stochastic BIM in some models suggests that the protection against adversarial attacks might not be as strong as with traditional BIM. Therefore, the choice between these methods would depend on the specific application requirements: whether maintaining higher clean accuracy or maximizing adversarial robustness is more critical.

6.5 DDPM Adversarial Training

The data and discussion presented highlight the significant impact of DDPM adversarial training on various deep learning models, showcasing distinct trends between CNN-based and transformer-based architectures. Specifically, CNN models such as VGG11_bn, ResNet 18, and Wide ResNet 50 experienced severe reductions in both clean and adversarial accuracy following DDPM training. These reductions, which approximate 75% to 83% in clean accuracy and 65% to 70% in adversarial accuracy, effectively reduce the performance of these models to near-random guessing levels.

In contrast, transformer models such as ViT_B_16, Swin_t, and ConvNeXt demonstrated different responses to the same training regimen. ViT and Swin_t showed only minor decreases in clean accuracy (3% and 6%, respectively) but major declines in adversarial accuracy (74% and 72%, respectively). Notably, the ConvNeXt model not only avoided a decrease in clean accuracy but actually improved by 6%, alongside a relatively modest reduction in adversarial accuracy of 51%. This performance continues the seen trend of a notable resilience in ConvNeXt compared to other models, highlighting its unique response to adversarial training challenges. More than that, it was the only model that consistently reached a very high accuracy (100% on each run on DDPM data). The detailed changes in performance metrics underscore the differential impacts of DDPM adversarial training across these model architectures, with transformer models generally exhibiting greater robustness than their CNN counterparts.

Note that this is the first work to our knowledge that investigates adversarial examples generated via a DDPM fed into CNN and Vision Transformers to compare their robustness. There has been prior work in the adversarial robustness space comparing CNNs and Vision Transformers, as described in the end of the Related Works section, but none that investigate data from modern generative models like DDPMs.

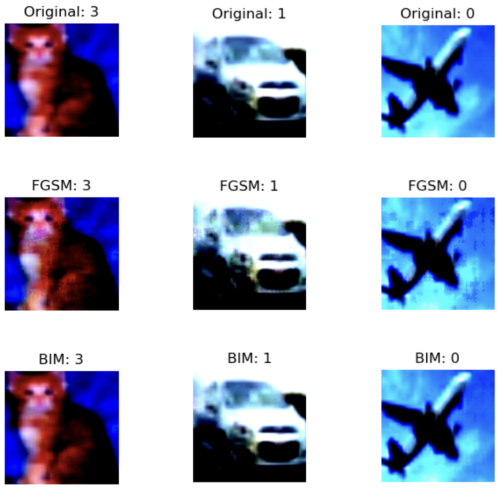

Figure 8: Clean, FGSM Perturbed and BIM Perturbed CIFAR-10 Images

6.6 Analysis

In the BIM experiments, it could be that 10 iterations of BIM training is perturbing the images outside of the data manifold. BIM, by its iterative nature, applies small but multiple perturbations to an image, progressively pushing it away from its original point in the input space and potentially to a region of the input space that the model has not encountered during standard training, effectively moving it "off-manifold." In practice, this can result in images that no longer resemble natural images or represent realistic scenarios, which challenges the model's ability to correctly interpret or classify them.

However, on manual inspection of the perturbed images versus the original CIFAR images, there isn't a overly noticeable perturbation in the images. Most of the BIM and FGSM perturbed images look akin to the classical adversarial example 4, where there may be a small amount of Gaussian noise visible on the perturbed image but not enough to be certain.

To further illustrate this, we investigated three random images from the training set, perturbed on FGSM and BIM on a random model (selected to be Swin_t) showed in Figure 8.

In the FGSM perturbed images, it's clear that there is a much more pronounced noise or blur and they are a bit discolored in certain parts of the image. Whereas in the BIM perturbed images, there is not many noticeable instances of blur or discolorations, with images being very similar to the original image. This follows the findings of our experiments well: BIM is much more expensive but prone to fooling the models. On the other hand, FGSM is cheap, but models have an easier time seeing through the obvious perturbations.

7 LIMITATIONS

Due to the complexity of the BIM attack, training occurred very slowly taking approximately five times as long as FGSM. A single epoch of BIM training consumed a a half hour of time whereas an entire ten epoch regime for FGSM training took only an hour. This severely

Note that our experimental scope was quite limited by the resources available for us to train and experiment with models, especially for computationally expensive attack techniques. For example, the Carlini Wagner (C&W) attack uses an optimization-based approach, which involves carefully crafting adversarial inputs that maximize classification errors while maintaining perceptual similarity to the original inputs. [2]. This optimization process is iterative and involves calculating gradients repeatedly until convergence, which can be computationally intensive and time-consuming.

Initially, we wanted to incorporate a wide variety of these adversarial attacks, like the C&W attack, as well as a larger range of models (specifically more Vision Transformers). We wanted to incorporate the MaxVIT model, which is also a PyTorch pretrained Vision Transformer (like ViT and Swin), but it was impossible due to the memory constraints. Note that the Rivanna High-Performance Computing (HPC) system provided by UVA offers the following GPUs for JupyterLab instances: NVIDIA RTX2090, RTX 3080, V100, A100 and A6000. Thus, the majority of experiments were run on a NVIDIA A6000 as it has the largest memory capacity of 48 GB of GDDR6 memory, larger than the competitive 40 GB A100.

8 THREATS TO VALIDITY

Due to the limited scale of our experiment we have a number of threats to the validity of our results. Due to the lack of computational resources, we were unable to run multiple, repeated trials of our experiments which led to single trial results being displayed. With more resources, we would be able to attempt a greater number of trials and be able to make claims about statistical significance. Additionally, we had limitations in regards to our computer vision models, our adversarial attacks, and our data generation models.

8.1 Computer Vision Models

Due to computational limitations, we were only able to run tests on six models to completion. Although we selected models from across nearly a decade of computer vision research, we still have gaps in the number and size of our models. Specifically due to GPU RAM limitations, we are only able to run tests on the smallest size of our models. Our experiments should be conducted across both a number of models and across various *sizes* of those models. The lack of cross-size evaluation represents a significant threat to the validity of our results. Larger models may demonstrate greater adversarial robustness over their smaller counterparts.

8.2 Adversarial Attacks

Within our attack methods, we only use two adversarial attacks. Further, neither attacks are black box attacks. The lack further exploration of different attacks such as Jacobian-Based Saliency Map attack, Universal Adversarial Attacks, and others, poses an additional threat to the validity of our results. Further, the lack of black box adversarial attacks such as Carlini and Wagner attacks and Natural Evolutionary Strategies attacks further poses a threat to the validity of our results. Incorporating a greater number of attacks as well as black box attacks could increase the validity of our results.

8.3 Data Generation Models

Within our experiments, we only use a single data generation model to improve adversarial robustness. But, there exist additional generative models such as EDMs and Adversarial Transformation Networks, the latter of which directly produces adversarial examples via generation. Further, the lack of conditionally generative models and the use of pseudo-labels poses a further attack on our validity.

9 CONCLUSION

In this paper, we tested adversarial attacks and a number of training methods against these attacks including training on the adversarially perturbed data, stochastic training, and training on generated data. Throughout these experiments, we have been able to see the robustness of advanced models such as ConvNeXt and Swin_t both of which employ architectures that focus on large-scale image features to classify images instead of the low-scale features that traditional CNNs rely on. This indicates that computer vision model research is progressing in the right direction: improvements in model architecture correlate with improvements in adversarial robustness.

Future areas of research include:

- How well do adversarial examples that are effective on one model generated by one method transfer and generalize to other models?
- Does the training on transferred examples improve the model's adversarial performance?
- How do different attack methods vary in transferability?

While answering these questions may improve the field's understanding of adversarial attacks, they also may improve the fields ability to conceptualize and develop novel computer vision architectures that display both improved image classification accuracy and adversarial robustness.

REFERENCES

[1] Y. Bengio, P. Simard, and P. Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks* 5, 2 (1994), 157–166. https://doi.org/10.1109/72.279181

[2] Nicholas Carlini and David Wagner. 2017. Towards Evaluating the Robustness of Neural Networks. arXiv:1608.04644 [cs.CR]

[3] Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, Percy Liang, and John C. Duchi. 2022. Unlabeled Data Improves Adversarial Robustness. arXiv:1905.13736 [stat.ML]

[4] T Ciodaro, D Deva, J M De Seixas, and D Damazio. 2012. Online particle detection with Neural Networks based on topological calorimetry information. *Journal of Physics: Conference Series* 368 (June 2012), 012030. https://doi.org/10.1088/1742-6596/368/1/012030

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. arXiv:2010.11929 [cs.CV]

[6] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2013. Rich feature hierarchies for accurate object detection and semantic segmentation. (2013). https://doi.org/10.48550/ARXIV.1311.2524 Publisher: [object Object] Version Number: 5.

[7] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International*

Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research, Vol. 9), Yee Whye Teh and Mike Titterington (Eds.). PMLR, Chia Laguna Resort, Sardinia, Italy, 249–256. https://proceedings.mlr.press/v9/glorot10a.html

[8] Ian J. Goodfellow, Yaroslav Bulatov, Julian Ibarz, Sacha Arnoud, and Vinay Shet. 2013. Multi-digit Number Recognition from Street View Imagery using Deep Convolutional Neural Networks. (2013). https://doi.org/10.48550/ARXIV.1312.6082 Publisher: [object Object] Version Number: 4.

[9] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. arXiv:1412.6572 [stat.ML]

[10] Sven Gowal, Sylvestre-Alvise Rebuffi, Olivia Wiles, Florian Stimberg, Dan Andrei Calian, and Timothy Mann. 2021. Improving Robustness using Generated Data. arXiv:2110.09468 [cs.LG]

[11] Ali Hatamizadeh, Jiaming Song, Guilin Liu, Jan Kautz, and Arash Vahdat. 2024. DiffiT: Diffusion Vision Transformers for Image Generation. arXiv:2312.02139 [cs.CV]

[12] Kaiming He and Jian Sun. 2014. Convolutional Neural Networks at Constrained Time Cost. (2014). https://doi.org/10.48550/ARXIV.1412.1710 Publisher: [object Object] Version Number: 1.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2014. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. (2014). https://doi.org/10.48550/ARXIV.1406.4729 Publisher: [object Object] Version Number: 4.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. (2015). https://doi.org/10.48550/ARXIV.1512.03385 Publisher: [object Object] Version Number: 1.

[15] Moritz Helmstaedter, Kevin L. Briggman, Srinivas C. Turaga, Viren Jain, H. Sebastian Seung, and Winfried Denk. 2013. Connectomic reconstruction of the inner plexiform layer in the mouse retina. *Nature* 500, 7461 (Aug. 2013), 168–174. https://doi.org/10.1038/nature12346

[16] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising Diffusion Probabilistic Models. arXiv:2006.11239 [cs.LG]

[17] C. Jarzynski. 1997. Equilibrium free-energy differences from nonequilibrium measurements: A master-equation approach. *Physical Review E* 56, 5 (Nov. 1997), 5018–5035. https://doi.org/10.1103/physreve.56.5018

[18] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. 2022. Elucidating the Design Space of Diffusion-Based Generative Models. arXiv:2206.00364 [cs.CV]

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger (Eds.), Vol. 25. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[20] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2016. Adversarial Machine Learning at Scale. (2016). https://doi.org/10.48550/ARXIV.1611.01236 Publisher: [object Object] Version Number: 2.

[21] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2017. Adversarial examples in the physical world. arXiv:1607.02533 [cs.CV]

[22] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. arXiv:2103.14030 [cs.CV]

[23] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. 2022. A ConvNet for the 2020s. arXiv:2201.03545 [cs.CV]

[24] Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2014. Fully Convolutional Networks for Semantic Segmentation. (2014). https://doi.org/10.48550/ARXIV.1411.4038 Publisher: [object Object] Version Number: 2.

[25] Zeyu Lu, Zidong Wang, Di Huang, Chengyue Wu, Xihui Liu, Wanli Ouyang, and Lei Bai. 2024. FiT: Flexible Vision Transformer for Diffusion Model. arXiv:2402.12376 [cs.CV]

[26] Junshui Ma, Robert P. Sheridan, Andy Liaw, George E. Dahl, and Vladimir Svetnik. 2015. Deep Neural Nets as a Method for Quantitative Structure–Activity Relationships. *Journal of Chemical Information and Modeling* 55, 2 (Feb. 2015), 263–274. https://doi.org/10.1021/ci500747n

[27] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2019. Towards Deep Learning Models Resistant to Adversarial Attacks. arXiv:1706.06083 [stat.ML]

[28] Kaleel Mahmood, Rigel Mahmood, and Marten van Dijk. 2021. On the Robustness of Vision Transformers to Adversarial Examples. arXiv:2104.02610 [cs.CV]

[29] Amir Najafi, Shin ichi Maeda, Masanori Koyama, and Takeru Miyato. 2019. Robustness to Adversarial Perturbations in Learning from Incomplete Data. arXiv:1905.13021 [stat.ML]

[30] Sylvestre-Alvise Rebuffi, Sven Gowal, Dan A. Calian, Florian Stimberg, Olivia Wiles, and Timothy Mann. 2021. Data Augmentation Can Improve Robustness. arXiv:2111.05328 [cs.CV]

[31] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. (2015). https://doi.org/10.48550/ARXIV.1506.01497 Publisher: [object Object] Version Number: 3.

[32] Leslie Rice, Eric Wong, and J. Zico Kolter. 2020. Overfitting in adversarially robust deep learning. arXiv:2002.11569 [cs.LG]

[33] Rulin Shao, Zhouxing Shi, Jinfeng Yi, Pin-Yu Chen, and Cho-Jui Hsieh. 2022. On the Adversarial Robustness of Vision Transformers. arXiv:2103.15670 [cs.CV]

[34] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. arXiv:1503.03585 [cs.LG]

[35] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Highway Networks. (2015). https://doi.org/10.48550/ARXIV.1505.00387 Publisher: [object Object] Version Number: 2.

[36] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. (2013). https://doi.org/10.48550/ARXIV.1312.6199 Publisher: [object Object] Version Number: 4.

[37] Jonathan Uesato, Jean-Baptiste Alayrac, Po-Sen Huang, Robert Stanforth, Alhussein Fawzi, and Pushmeet Kohli. 2019. Are Labels Required for Improving Adversarial Robustness? arXiv:1905.13725 [cs.LG]

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. arXiv:1706.03762 [cs.CL]

[39] Zekai Wang, Tianyu Pang, Chao Du, Min Lin, Weiwei Liu, and Shuicheng Yan. 2023. Better Diffusion Models Further Improve Adversarial Training. (2023). https://doi.org/10.48550/ARXIV.2302.04638 Publisher: [object Object] Version Number: 2.

[40] Hui Y. Xiong, Babak Alipanahi, Leo J. Lee, Hannes Bretschneider, Daniele Merico, Ryan K. C. Yuen, Yimin Hua, Serge Gueroussov, Hamed S. Najafabadi, Timothy R. Hughes, Quaid Morris, Yoseph Barash, Adrian R. Krainer, Nebojsa Jojic, Stephen W. Scherer, Benjamin J. Blencowe, and Brendan J. Frey. 2015. The human splicing code reveals new insights into the genetic determinants of disease. *Science* 347, 6218 (Jan. 2015), 1254806. https://doi.org/10.1126/science.1254806

[41] Runtian Zhai, Tianle Cai, Di He, Chen Dan, Kun He, John Hopcroft, and Liwei Wang. 2019. Adversarially Robust Generalization Just Requires More Unlabeled Data. arXiv:1906.00555 [cs.LG]

[42] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. 2019. Theoretically Principled Trade-off between Robustness and Accuracy. arXiv:1901.08573 [cs.LG]

Received 4 May 2024