Preliminary Studies on Defending Image Adversarial Attacks with Domain Adaptation Algorithms

Zheng Zhang

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2023-72
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2023/EECS-2023-72.html

May 8, 2023

Preliminary Studies on Defending Image Adversarial Attacks with Domain Adaptation Algorithms

by Zheng Zhang

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

Committee:

Professor Alberto L. Sangiovanni-
Vincentelli
Research Advisor

April 26, 2023

(Date)

* * * * * * *

Professor Jiantao Jiao
Second Reader

5/1/23

(Date)

Preliminary Studies on Defending Image Adversarial Attacks with Domain Adaptation Algorithms

by

Zheng Zhang

A thesis submitted in partial satisfaction of the

requirements for the degree of

Master's of Science

in

Electrical Engineering and Computer Sciences

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Alberto L. Sangiovanni-Vincentelli, Chair
Professor Jiantao Jiao

Spring 2023

Abstract

Preliminary Studies on Defending Image Adversarial Attacks with Domain Adaptation Algorithms

by

Zheng Zhang

Master's of Science in Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Alberto L. Sangiovanni-Vincentelli, Chair

Deep Neural Networks (DNNs) have demonstrated high performance in various tasks using image datasets. Despite the rapid expansion of innovation and research in DNNs, they are also vulnerable to image-based adversarial attacks, which can compromise the reliability of DNNs and impose challenges on the applications of artificial intelligence (AI) in safety-critical tasks. In this report, we propose a new defense method that takes advantage of the domain loss function of Domain Adaptation Algorithms, and we have named the method Domain Adaptation Defense (DAD). DAD can generate distributional-based defense without prior knowledge of attack functions, making it more applicable in real-life applications. Our results also indicate that DAD can perform similarly to many current defense methods. Through our study of distributional discrepancies, we verify that the domain loss function is an essential defense mechanism that captures the domain differences between clean and adversarial images. From the comparison results, we identify that existing Domain Adaptation Algorithms with domain-classifier-based loss functions, such as Proxy A-Distance, are more effective than the others. Furthermore, we have designed a new experimental procedure for studying the joint research area between distributional shifts of adversarial attacks and Domain Adaptation Algorithms. The promising results and well-formatted procedure will inspire improvements and inventions of new domain loss functions and Domain Adaptation Algorithms focusing on defending against adversarial attacks.

Keywords: Domain Adaptation, Defend Adversarial Attacks, Adversarial Training, Domain Distributional Shift, Domain Loss Functions

Acknowledgments

This research report concludes the end of my five-year academic journey at UC Berkeley and eight years of education in the U.S. since I left my home country for better opportunities. I am incredibly grateful to the many people who provided a tremendous amount of support along this challenging yet exciting journey. I would like to take a moment to express my sincere gratitude to my research advisor, Professor Alberto Sangiovanni-Vincentelli, for offering this fantastic research opportunity and creating a supportive and encouraging environment for me to learn and grow in the world of AI. Your unconditional and patient support helped me find my passion for research and learn to communicate academically and professionally. I want to thank my research mentors, Xiangyu Yue and Baihong Jin, for introducing me to the world of AI research and being the lighthouse in my research career, helping me find my passion and direction. Graduating during a tech recession is not a fun place to be, but thanks to my family's support and love, I overcame life's obstacles and came out stronger. Lastly, I thank many others who have provided help along the way because none of my achievements would have been possible without them.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

The evolution of Deep Neural Networks (DNNs) has demonstrated considerable success by mastering tasks such as image classification, segmentation, and object localization and detection in the past decade. For example, the state-of-the-art image classification DNNs on ImageNet have evolved from AlexNet with 63% accuracy in 2012 [16], to ResNet-152 with 78% accuracy in 2015 [12], and to Finetuned-CoCa with 91% accuracy in 2022 [25]. These unparalleled breakthroughs also lead to extensive applications of DNNs in advanced tasks such as video recognition and autonomous driving. However, a significant concern in extending DNNs to safety-critical tasks is their vulnerability to image-based adversarial attacks. Simple adversarial attack methods, such as the Black-box One Pixel Attack, can compromise the stability and accuracy of various DNNs.

While most of the current defense methods against adversarial attacks focus on data augmentation, adversarial training, and identifying the attack functions, we propose a new and creative approach to convert an adversarial defense task into a cross-domain classification task using Domain Adaptation Algorithms [18]. In other words, we set up our hypothesis that an image adversarial attack is a form of domain distributional shift of the clean dataset, and we will validate our hypothesis with experiments. As far as we know, this is one of the first research studies to approach defense methods focusing on the distributional shift of image-based adversarial attacks.

1.2 Motivation

Reflections on Existing Defense Methods

Many works have been conducted to defend against adversarial attacks and improve the stability and robustness of DNNs. The current adversarial defense methods can be roughly summarized into the following categories [18]:

- **Adversarial Training:** Improves the accuracy and robustness of DNNs against adversarial attacks by adding adversarial examples to the training set. If the model has seen the adversarial examples during training, it is more likely to make correct classifications when facing a test set mixed with adversarial examples [23].
- **Data Augmentation:** Applies a mix of randomization and image augmentation methods, such as rotation, random cut, and normalization, to reduce the effect of noise added to clean images by adversarial attacks [26].
- **Attack Function Based Defense:** Most of the attacks are generated from well-defined attack functions or procedures. Some defense methods take attack functions as given and incorporate them into the model's loss calculation so that the subjected models optimize against the attack functions in the training process [18].

We propose a new defense method, Domain Adaptation Defense (DAD), that provides the same level of defense performance without knowing the actual attack functions. This makes DAD an effective and attack-function-agnostic method applicable to defend against any attacks that shift domain distributions. An attack-function-agnostic method is more practical in real-life applications because the attackers will not reveal the attack functions. The key difference in our approach is to focus on the distributional shift aspect of adversarial attacks and quantify such shift with an appropriate method to incorporate it into the loss function.

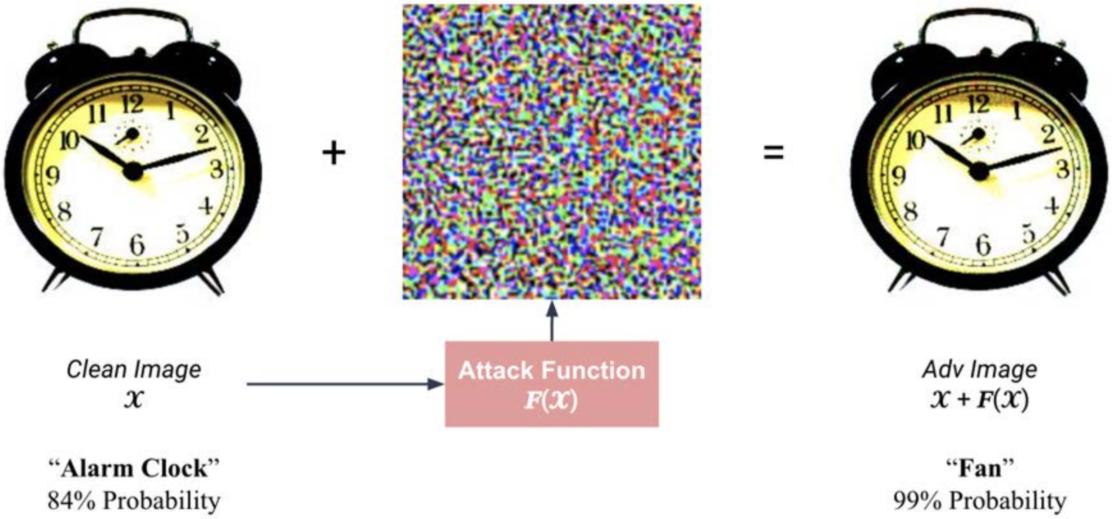


Figure 1.1: An example of a famous image adversarial attack method is the Fast Gradient Signed Method (FGSM) attack. A noise generated from the sign attack function is added to the original panda image, which confuses the model to make the wrong prediction [8].

Connecting Distributional Shifts and Adversarial Attacks

Figure 1.1 demonstrates an example of FGSM adversarial attack. For a clean image, a regular DNN model can make the correct prediction with an 84% probability (ResNet-50 in our case). However, when we add a perturbation generated by the FGSM attack function to the clean image, the DNN model becomes extremely confident about the wrong prediction. In other works, FGSM attack completely "fooled" the targeted DNN model. In addition to the impact on accuracy, we focused on the noise generated from the attack. In particular, the perturbation noise represents a shift from the original image pixel distribution. This realization makes us wonder if there could be a connection between adversarial attacks and image distributional shifts.

Classification on Domains with Different Distributions

The benefit of connecting distributional shifts and adversarial attacks is that it allows us to use existing tools that specialize in cross-domain classification. Domain Adaptation Algorithms are well-studied methods that specialize in making supervised and unsupervised predictions on datasets with shifted domains [3]. Then, under the setup of a Domain Adaptation task, we can think of a clean image dataset as the *Source Domain*, and the adversarial image dataset as the *Target Domain*.

1.3 Related Works

Generate Adversarial Examples

In this report, we generate a large number of adversarial examples for experimentation. We obtain a general idea of the current state-of-the-art image adversarial attacks from Han's work for an overview of the methods [24]. Han's work provides a great introduction to classic attack methods and their effectiveness in reducing the classification accuracy of DNNs. To understand how each type of adversarial attack is generated, we uncover the mathematical intuitions by referring to the original papers [8][11][17][21][21][11]. In addition to theoretical understandings of adversarial attacks, we utilized a great open-source tool for generating the adversarial datasets in PyTorch environment [15].

Measurement of Distributional Shift of Adversarial Examples

We experiment with two common methods for measuring distributional shift: Maximum Mean Discrepancy (MMD) and Proxy A-Distance (PAD). We studied the theoretical intuitions behind them and conducted experiments with adversarial datasets. For MMD, we refer the original paper and other related papers for algorithms and explanations [27][6][9][10]. For PAD, we also refer to the first paper that proposed the measurement by Ben-David and the following works for approximations [1][7]. Furthermore, our literature review indicates a natural disadvantage of MMD compared to PAD in terms of their ability to detect adversarial attacks, and we refer to Nicholas' work to generate experiment ideas [2].

Domain Adaptation Algorithms Selection

The critical part of the paper is to defend against adversarial attacks with Domain Adaptation Algorithms. Thus, we acquire knowledge of the definitions, domain setup, experimentation, and benchmark performance from Yao's work on a general summary of Domain Adaptation methods [18]. With the correct benchmark, we carefully selected three algorithms: DANN, JAN, and CDAN, for experiments. For benchmark information, we refer to the original papers that were evaluated under the OfficeHome dataset [4][20][19][13][14]. To understand why Domain Adaptation Algorithms can defend against adversarial attacks, we looked at the mathematical definitions of the loss functions [5][19][20][13]. Furthermore, we also refer to a great open-source codebase to train DNNs with selected Domain Adaptation Algorithms [14][13].

1.4 Organization

The following chapters of this report are organized as the followings:

- **Chapter 2: Methodology** This chapter provides insights on how to conduct adversarial attacks, measure distribution shift, and explanations of domain adaptation algorithms

- **Chapter 3: Experiment** This chapter lists out the experimental design and approaches, and we defines metrics for comparisons.

- **Chapter 4: Results** This chapter reports the experiment results and the main findings.

- **Chapter 5: Analysis and Discussions** This chapter analyzes experiment results and provides explanations.

- **Chapter 6: Extensions** We extend the same approach to DNNs with less complexity in this section to ensure that our method is the application to other DNNs.

- **Chapter 7: Conclusion** We conclude the main findings, outline the contributions, and provide future research directions.

Chapter 2

Methodology

2.1 Overview

To explain how we approach the study, we list the data, algorithms, measurement methods, and models used during the experiment. In addition, this section provides the theoretical insights behind the algorithms used to generate different kinds of adversarial attacks and Domain Adaptations from the original papers.

2.2 Data

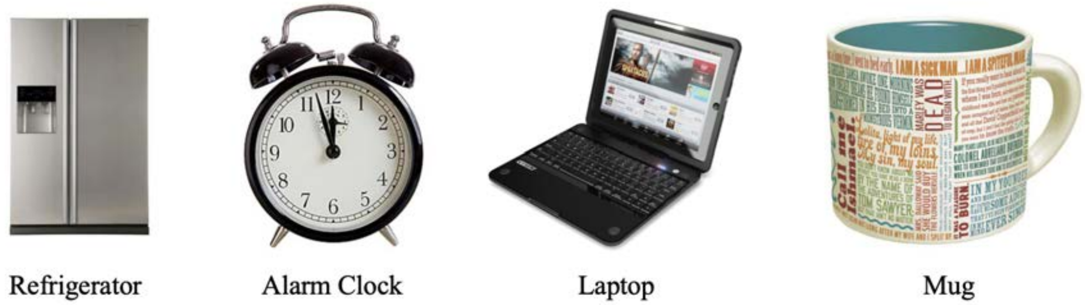


Figure 2.1: A sample of the *Real-World* class images from the *Office-Home* dataset [22].

As we are using Domain Adaptation (DA) as a defense method, we aim to employ a well-known dataset for benchmarking DA algorithms. This will allow for fair comparisons and enable us to select the best set of DA models. We have chosen to use the *Office-Home* dataset, which is widely used in DA research and benchmarking [22]. Specifically, we are focusing on using the *Real-World* portion of the dataset since most adversarial research utilizes images from the real world, such as the ImageNet dataset. The assumption here is that *Real-World* portion of the dataset have similar sample distribution as the ImageNet dataset.

2.3 Methods for Generating Adversarial Examples

Adversarial attack methods are algorithms that compromise the reliability and accuracy of DNNs by adding perturbations to testing or out-of-sample data. Generally, there are two major categories of attacks depending on the adversary's knowledge: White-Box and Black-Box attacks. White-Box attacks have access to all information, including the targeted model parameters, architecture, optimization functions, and gradients. On the other hand, Black-Box attacks have no information or access to model information, often resulting in weaker attacks that are more computationally easy to execute [24]. Additionally, it is easier to adjust attack strengths for White-Box attacks compared to Black-Box attacks. Given that White-Box attacks are well-researched and more effective than Black-Box models, our research focuses on attacking the Office-Home dataset with well-known gradient-based White-Box attacks. The gradient-based White-Box attack methods that we use are the Fast Gradient Sign Method (FGSM), Projected Gradient Descent (PGD), and Basic Iterative Method (BIM).

Fast Gradient Sign Method (FGSM)

Fast Gradient Sign Method (FGSM) is a simple and effective attack method that leverages the knowledge of the targeted model's loss function to generate gradient attack [8]. Mathematically, FGSM's attack function is defined as:

$$X_{adv\_image} = X_{clean\_image} + \epsilon \cdot \text{sign}\left(\nabla_X \mathcal{L}(X_{clean\_image}, y)\right) \tag{2.1}$$

where X_{clean_image} is the original clean image and X_{adv_image} is the adversarial image. y is the true label of the clean image. Specifically, FGSM attacks clean images by computing the sign of the gradient of the loss value and adding a cosntant strength, $\epsilon \in [0, 1]$, to the clean image. The larger the ϵ, the stronger the attack. As simple as the attack function, FGSM is one of the most effective and computationally easy ways to generate adversarial examples, which makes it an ideal benchmark for measuring the robustness of DNNs [11].

Basic Iterative Method (BIM)

BIM method is the iterative version of the FGSM attack. BIM takes smaller steps in each iteration to generate adversarial attacks. Mathematically, the BIM's attack function is defined as [17]:

$$\boldsymbol{X}_0^{adv\_image} = \boldsymbol{X_{clean\_image}}$$

$$\boldsymbol{X}_{N+1}^{adv\_image} = \mathrm{Clip}_{X,\epsilon}\left\{\boldsymbol{X}_{N}^{adv\_image} + \alpha\,\mathrm{sign}\left(\nabla_X \mathcal{L}\left(\boldsymbol{X}_{N}^{adv\_image}, y\right)\right)\right\} \quad (2.2)$$

where BIM starts with clean image X_{clean_image} in time step 0, then changing the adversarial image by evolving the results from the Clip function at each time step. α is the value change on each pixel on each time step. The Clip function restricts inputs into a constraint of $\|X - X'_N\|_\infty \leq \epsilon$ [11]. The larger the constraint, the stronger the attack strength.

Projected Gradient Descent (PGD)

PGD is also an iterative gradient-based attack method. It is similar to BIM except that PGD initializes the adversarial examples from a randomly generated position. The effect of a random start point is similar to adding a random noise at time step 0 to the clean image. Mathematically, PGD's attack function is defined as[21]:

$$\boldsymbol{X}_{0}^{adv\_image} = \boldsymbol{X_{clean\_image}} + S$$

$$\boldsymbol{X}_{N+1}^{adv\_image} = \mathrm{Clip}_{X,\epsilon}\left\{\boldsymbol{X}_{N}^{adv\_image} + \alpha\,\mathrm{sign}\left(\nabla_X \mathcal{L}\left(\boldsymbol{X}_{N}^{adv\_image}, y\right)\right)\right\} \quad (2.3)$$

where every parameters are the same as BIM, except for S, which is a randomly generated start point with $\|S\|_\infty \leq \epsilon$ [21][11].

2.4 Image Data Distribution Shift Measurement Methods

While it is easy to observe that adversarial attacks cause shifts in image distribution by adding noise to the original pixels, we aim to find methods to quantify the distribution shift caused by adversarial attacks. Understanding the degree of distributional shift enables correlation analysis between adversarial attacks and model performance. Through literature review, we have identified two existing methods for measuring distributional discrepancy between two datasets: Maximum Mean Discrepancy (MMD) and Proxy A-Distance (PAD). In this section, we will formally define the two measurement methods and compare their applications on adversarial examples.

Maximum Mean Discrepancy (MMD)

Maximum Mean Discrepancy (MMD) is a widely used method for measuring distributional discrepancy between two data sources [27]. Formally, let P and Q be the two data sources, and let μ_P and μ_Q be the kernal means embeddings of the two dataset. In the case of image data, kernel means embeddings can be generated from activation functions in DNNs. MMD between two datasets is defined as [6]:

$$\mathrm{MMD}(P, Q; \mathcal{F}) = \sup_{\|f\| \leq 1} |E[f(X)] - E[f(Y)]| \quad (2.4)$$

in which f is a continuous function from some kernel space such as multi-scale and rbf. One famous example of such kernel, proposed by Gretton's work, belonging to a unit ball in the *reproducing kernel Hilbert space* (RKHS) [9]. Therefore, under Gretton's suggestion, we can future write the definition of MMD as:

$$\text{MMD}(P, Q; \mathcal{F}) = \|\mu_P - \mu_Q\|_{\mathcal{H}_k} \tag{2.5}$$

in which k is restricted by the RKHS kernel [10].

Proxy A-Distance (PAD)

PAD provides measurements of discrepancy between two probability distributions. In the context of image dataset, we can think of probability distribution as the distribution of pixels in each color channel across all images in a dataset. While there is a formal probability definition of A-Distance introduced in Ben-David's work [1], a more intuitive way to obtain a proxy of the A-Distance was introduced by Xavier [7]:

1. Suppose we have two distribution domains: P and Q
2. Randomly shuffle and mix P and Q into M, and label each image as P or Q (so that M only has two classes)
3. Perform train-test split on M
4. Train a simple binary classifier, such as SVM, on train set to predict the domain of each image (e.g. to predict if an image is from domain P or domain Q)
5. Compute error (ϵ) on the hold-out test set
6. Calculate PAD $= 2(1 - 2\epsilon)$

The main idea is that if the distributional difference between the probability (or feature) distributions of P and Q is significant enough, any simple binary classifier will be able to accurately classify images into the two domains. This will result in high test accuracy and low error. On the other hand, if the difference between P and Q is too small, a simple binary classifier will not be able to classify the two domains. This will result in low test accuracy (around 50%) and high error because the classifier is randomly guessing the domain of each test image.

Application of MMD and PAD on Adversarial Examples

In theory, both MMD and PAD can measure distributional discrepancy between the original images and adversarial images, but our literature review and experiment results indicate that MMD is susceptible to adversarial attacks like DNNs, and thus not an ideal method in our context. Carlini Wagner empirically showed that MMD failed to recognize distributional

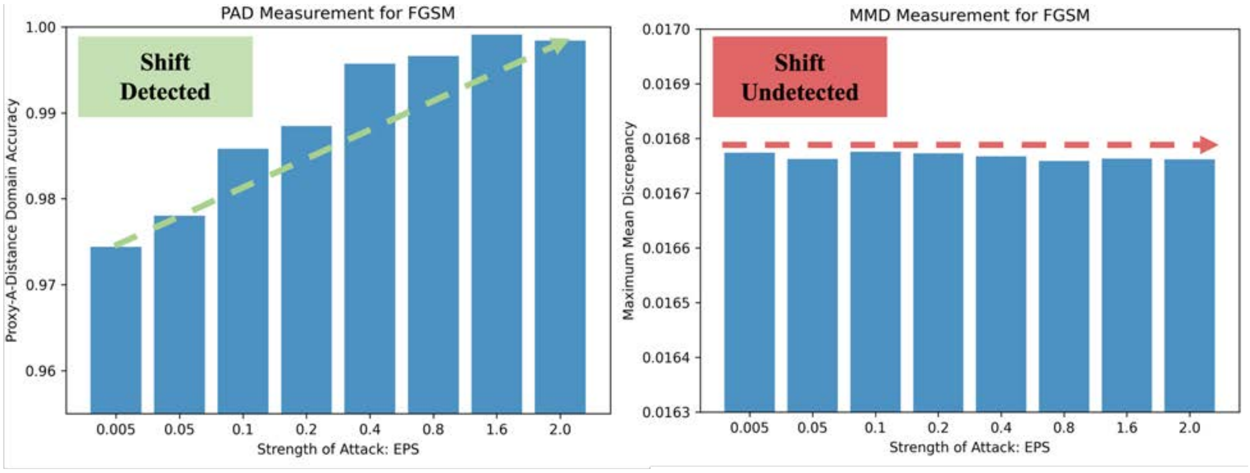


Figure 2.2: These two plots show how PAD and MMD measurements change as the attack strength of FGSM increases from left to right on the x-axis. For PAD (left), the y-axis shows the domain classifier accuracy on the test data. Higher accuracy indicates a larger shift in domain distributions. For MMD (right), the measurement barely changes across all levels of attack, indicating that MMD is incapable of detecting the domain shift caused by adversarial attacks.

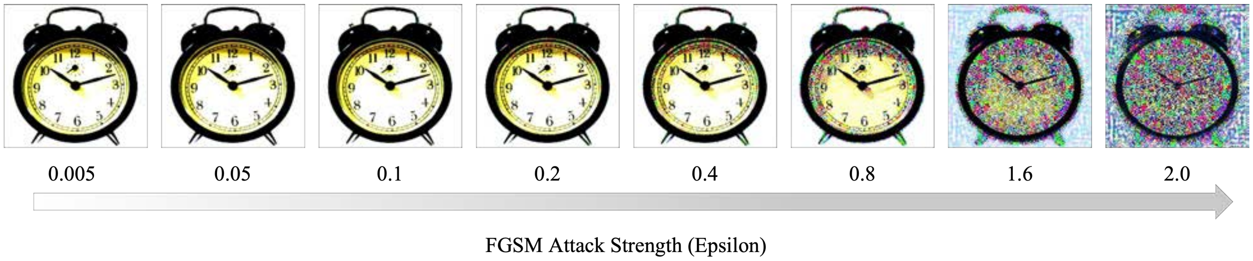


Figure 2.3: Use an image from the Alarm Clock class as an example. These plots visually show how image distribute changes as the strength of FGSM attack increases. It is easy to see that image starts to show noticeable changes from epsilon = 0.4, yet the measurements of MMD stay relatively the same across all attack strengths.

discrepancy between the original images and adversarial images [2].

Our experiment result in Figure 2.2 provides a direct comparison between PAD and MMD measurements under the same type of adversarial attack (FGSM) and across the same levels of attack strengths. For PAD (left), the y-axis displays the test accuracy of the domain binary classifier. Thus, the higher the test accuracy, the larger the shift in domain distribution. For MMD (right), the y-axis shows the actual MMD value computed with the rbf kernel. It is easy to see a strong and positive correlation between PAD measurement and the degree of distribution shift in dataset. However, for MMD, the measurement is about the same across all attack strengths, which indicates that MMD is incapable of detecting distribution shift under image adversarial attack. To further show the image distribution shift, Figure 2.3 provides a visualization on image distribution shift using an example from the Alarm Clock class. It is easy to see the effect of FGSM attack starting from epsilon = 0.4 and afterward. Therefore, we decided to use PAD for measuring domain distribution shift between the clean and adversarial images for experiments.

2.5 Domain Adaptation Algorithms and DNN Algorithm Selection

Domain Adaptation algorithms aim to provide accurate prediction on a target domain by addressing the domain distributional differences between the source and target domains. In our study, as show in Figure 2.4, we define clean images as the *Source Domain*, and adversarial images as the *Target Domain*. The goal is to defend adversarial attack by applying Domain Adaptation algorithms to make correct predictions on adversarial images.

We carefully select some of the most well-studied Domain Adaptation algorithms so that we get a fair compassion between models on benchmark performance. Based on algorithms performance on the *Office* benchmark from the original papers, we finalize a set of models of **DANN, JAN, and CDAN**. These models are preferred because:

1. Their benchmark performances are different from each other. From table 2.1, we can see that their performance are ranked as **CDAN** > **JAN** > **DANN**. This could help provide further insights on the relationship between Domain Adaptation's performance and the effectiveness of our defense method [4][20][19].

2. They use different domain loss functions to detect domain differences, which can help us understand why Domain Adaptation algorithms can be used for defending adversarial attacks [4][20][19]. If one type of loss function is less effective than the other, we can conduct root-cause analysis based on the differences.

In terms of DNN, we select **ResNet-50**, the most popular model in Domain Adaptation research, performance benchmark, and experiments [4][20][19]. To account for model size bias, we also study the performance of **ResNet-18** in Chapter 6 as an extension.

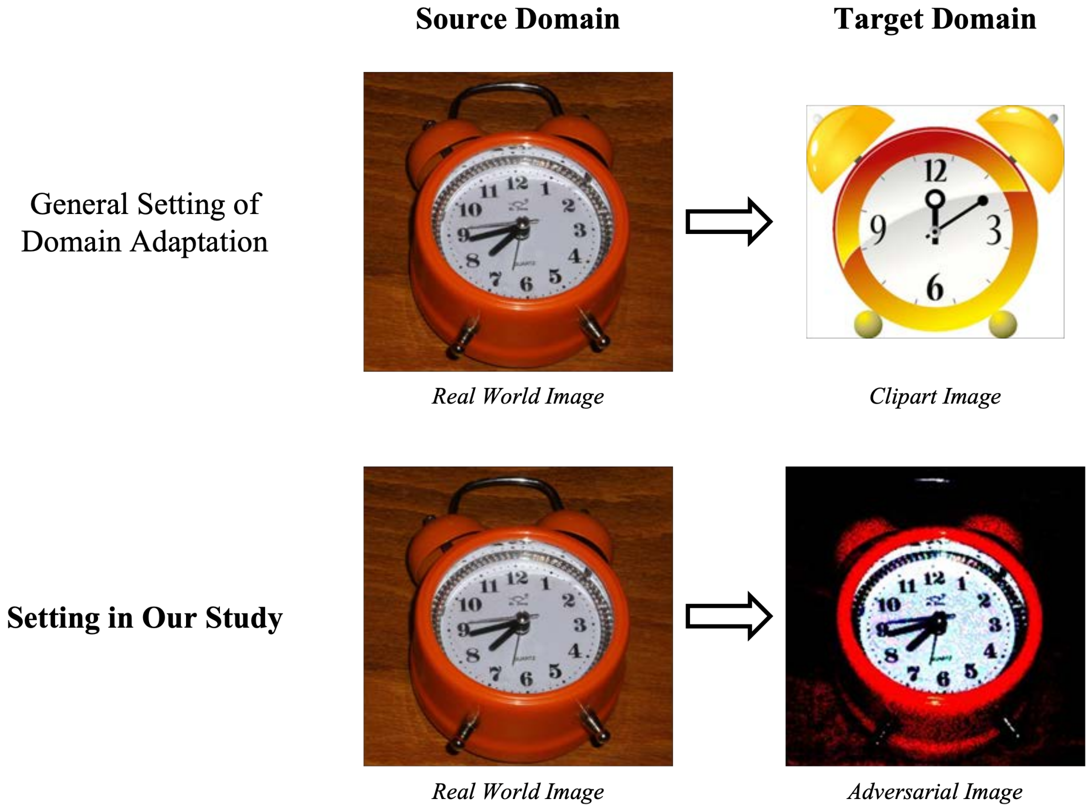


Figure 2.4: This figure provides a side-by-side comparison between the objectives of a general Domain Adaptation classification task and adversarial defense task in our study. The difference is that we think of adversarial examples as a shifted source domain, so we can treat them as the target domain defined by the conventional Domain Adaptation classification task.

Domain Adaptation Algorithm	Average Accuracy (across all domains)	Rw → Ar	Rw → CI	Rw → Pr
DANN	57.6%	71.1%	60.7%	81.1%
JAN	58.3%	72.5%	55.9%	80.5%
CDAN	65.8%	75.5%	61.5%	83.8%

Table 2.1: Benchmark results of the selected Domain Adaptation Algorithms [4][20][19]. These benchmark performances are evaluated on the *OfficeHome* dataset using the ResNet-50 DNN model. Domain Adaptation tasks are denoted as: Source → Target, following the convention of the *OfficeHome* dataset. The Average Accuracy is calculated on all possible Domain Adaptation pairs performed on the domains of the *OfficeHome* dataset [13][14].

Chapter 3

Experiment

3.1 Overview

This chapter covers the experimental design and setup in our study. Therefore, we define the setup, goals, and procedure of the experiment as follow:

Goal: Discover if Domain Adaptation Algorithm is an effective way to defend against image adversarial attacks. If so, find out the explanation behind the advantage of Domain Adaptation Algorithm.

Experiment Setup:

1. We use ResNet-50 as our baseline model.
2. We use three kinds of attack algorithms: Fast Gradient Sign Method (FGSM), Basic Iterative Method (BIM), Projected Gradient Descent (PGD).
3. For each attack algorithm, generate 5 different levels of attack strength to create more treatment and control groups for comparisons.
4. By the end of the experiment setup, we will have 15 adversarial datasets for experimentation.

Experiment Procedure:

1. Compute and record Proxy A-Distance domain shift measurements for each adversarial dataset.
2. Train a baseline model on the clean images with ResNet-50.
3. Record baseline model's accuracy on each adversarial dataset, and we define this set as Baseline accuracy.

4. Train ResNet-50 with Domain Adaptation Algorithms: DANN, JAN, and CDAN. We setup the training process with Clean Images as the *Source Domain* and Adversarial Images as the *Target Domain*. We will end up with 45 models trained with Domain Adaptation algorithms. Then, generate Domain Adaptation models' accuracy on adversarial dataset.

5. Compare Domain Adaptation accuracy with the Baseline accuracy, and discover the differences.

6. Analyze the relationship between the Proxy A-Distance domain shift and the accuracy differences.

The following subsections provide details of each step of the experiment.

3.2 Generate Adversarial Datasets

Fast Gradient Sign Method (FGSM)

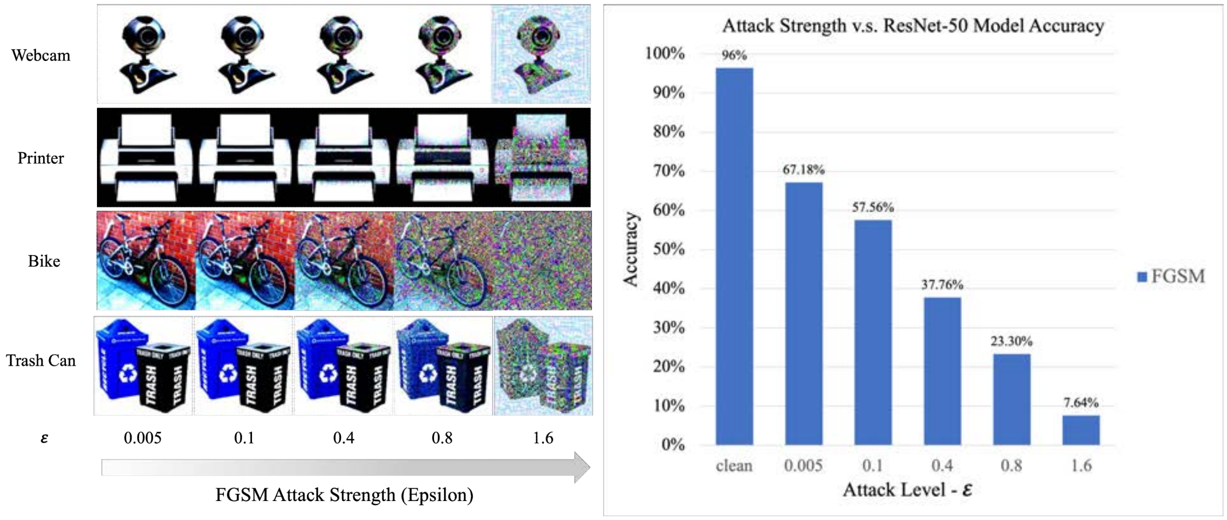


Figure 3.1: (Left) A sample of adversarial images created with FGSM attack. The x-axis indicates increasing attack strengths and y-axis provides the class labels. For FGSM attack, the larger the ϵ, the stronger the attack. (Right) FGSM attack on the ResNet-50 DNN model. As attack strength increases, the validation accuracy of ResNet-50 model decreases.

We generate adversarial examples with FGSM on 5 attack levels with $\epsilon = [0.005, 0.1, 0.4, 0.8, 1.6]$. We target a baseline model ResNet-50 for generating adversarial examples. Figure 3.1 provide a sample of visualizations from the datasets we created. The attack noise might be different across images because FGSM method creates a unique noise for each batch of data. Figure 3.1 also provides a visualization on how ResNet-50 accuracy decreases as the attack strength increases with FGSM.

Basic Iterative Method (BIM)

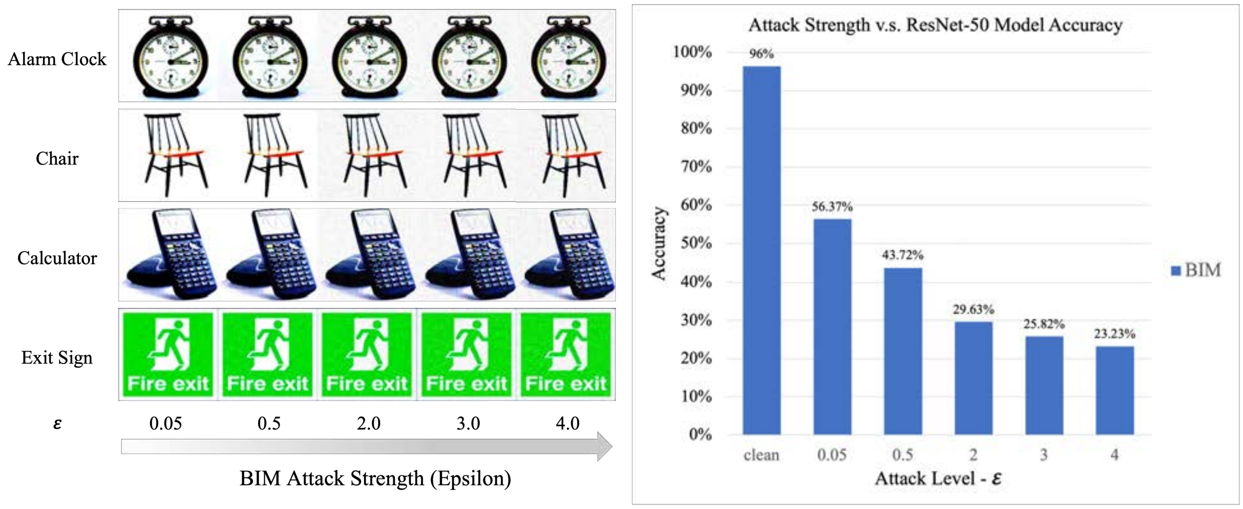


Figure 3.2: (Left) A sample of adversarial images created with BIM attack. The x-axis indicates increasing attack strengths and y-axis provides the class labels. For BIM attack, the larger the ϵ, the stronger the attack. (Right) BIM attack on the ResNet-50 DNN model. As attack strength increases, the validation accuracy of ResNet-50 model decreases.

Similar to the case of FGSM, we generate adversarial examples with BIM on 5 attack levels with ϵ = [0.05, 0.5, 2, 3, 4]. Since BIM is slightly different than FGSM, the same epsilon value does not produce the same drop in accuracy. Therefore, we picked different set of epsilon to differentiate attack strengths. Figure 3.2 provide a sample of visualizations from the datasets we created. The attack noise might be different across images because BIM method creates a unique noise for each batch of data. Figure 3.2 also provides a visualization on how ResNet-50 accuracy decreases as the attack strength increases with BIM.

Projected Gradient Descent (PGD)

We generate adversarial examples with PGD on 5 attack levels with ϵ = [0.05, 0.5, 2, 3, 4]. We target a baseline model ResNet-50 for generating the examples. Figure 3.3 provide a sample of visualizations from the datasets we created. The attack noise might be different across images because PGD method creates a unique noise for each batch of data. Figure

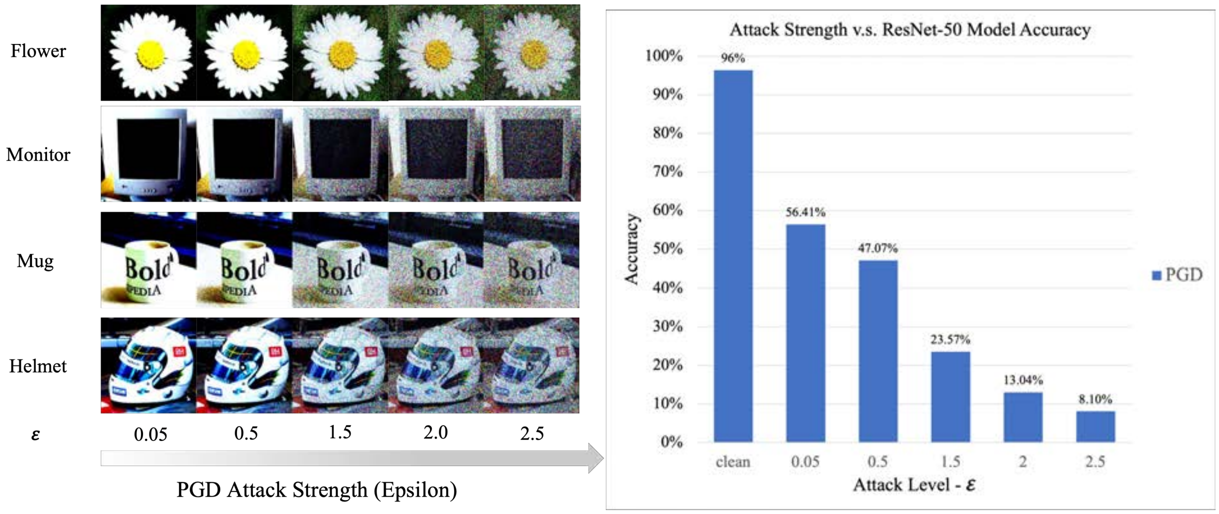


Figure 3.3: (Left) A sample of adversarial images created with PGD attack. The x-axis indicates increasing attack strengths and y-axis provides the class labels. For PGD attack, the larger the ϵ, the stronger the attack. (Right) PGD attack on the ResNet-50 DNN model. As attack strength increases, the validation accuracy of ResNet-50 model decreases.

3.3 also provides a visualization on how ResNet-50 accuracy decreases as the attack strength increases with PGD.

From the samples of images above, we see that the noise generated from all three methods are different. The adversarial datasets we have provide a good variance of attack on the baseline ResNet-50 models.

3.3 Proxy A-Distance Domain Shift Measurement

As discussed in Chapter 2.4, PAD is an effective method for detecting and measuring domain distributional shifts. Computing measurements on distribution shift is important because it allows us to analyze the relationship between the effectiveness of Domain Adaptation algorithms and the degree of shift in distribution. Furthermore, as far as we know, we are the first research in the area that provides PAD measurement on different adversarial attack methods. Therefore, this experiment is also a good introduction to the methodology of studying the distributional shift of image adversarial attacks. We compute the PAD in terms of the test accuracy of the domain classifier for all adversarial datasets from the previous section.

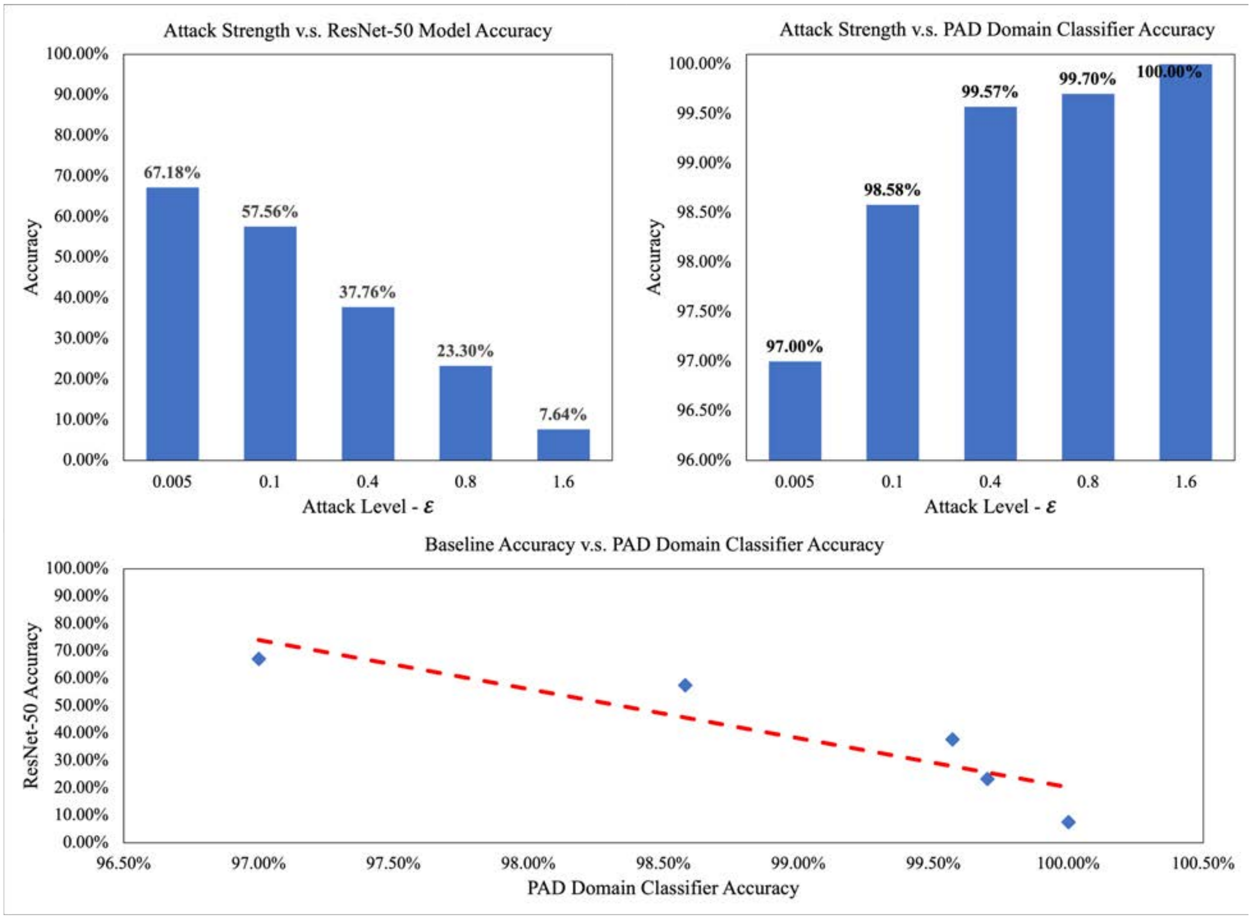


Figure 3.4: This plot indicates a positive relationship between attack strength and PAD measurement (a negative relationship between baseline model accuracy and PAD measurement). The change of PAD domain distributional shift measurements at each attack level validates that FGSM adversarial attack causes distributional shift from the original images.

Fast Gradient Sign Method (FGSM)

We summarized our calculation in Figure 3.4, in which we see a positive relationship between FGSM attack strength and PAD domain distributional shift measurement. That is, when the attack strength increases, PAD domain distributional difference measurement becomes more significant. These results verify that FGSM attack can be viewed as a form of change in dataset distribution, which is what Domain Adaptation Algorithms are specialized in solving.

Basic Iterative Method (BIM)

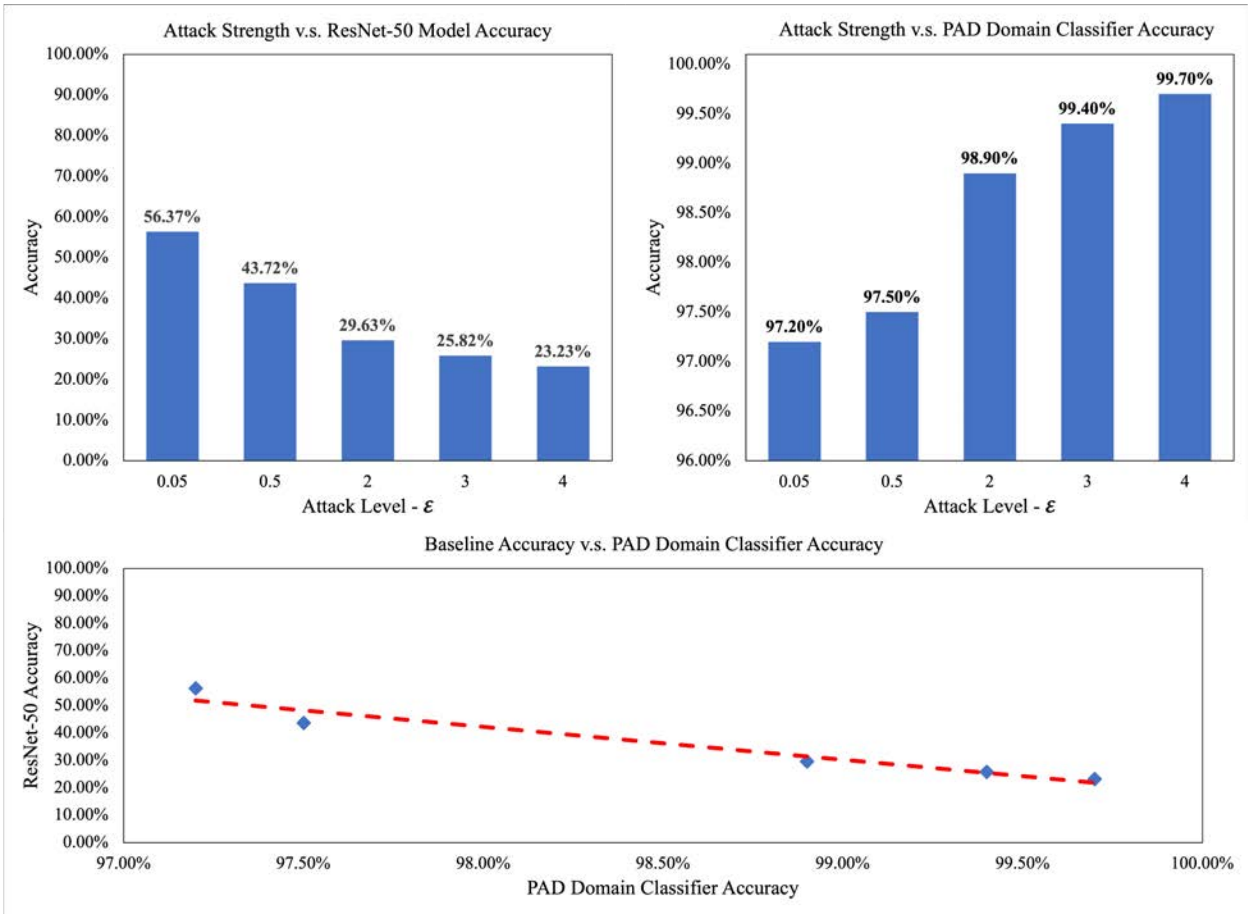


Figure 3.5: This plot indicates a positive relationship between attack strength and PAD measurement (a negative relationship between baseline model accuracy and PAD measurement). The change of PAD domain distributional shift measurements at each attack level validates that BIM adversarial attack causes distributional shift from the original images.

We summarized our calculation for BIM in Figure 3.5; Similar to the case of FGSM, we see a positive relationship between BIM attack strength and PAD domain distributional shift measurement.

Projected Gradient Descent (PGD)

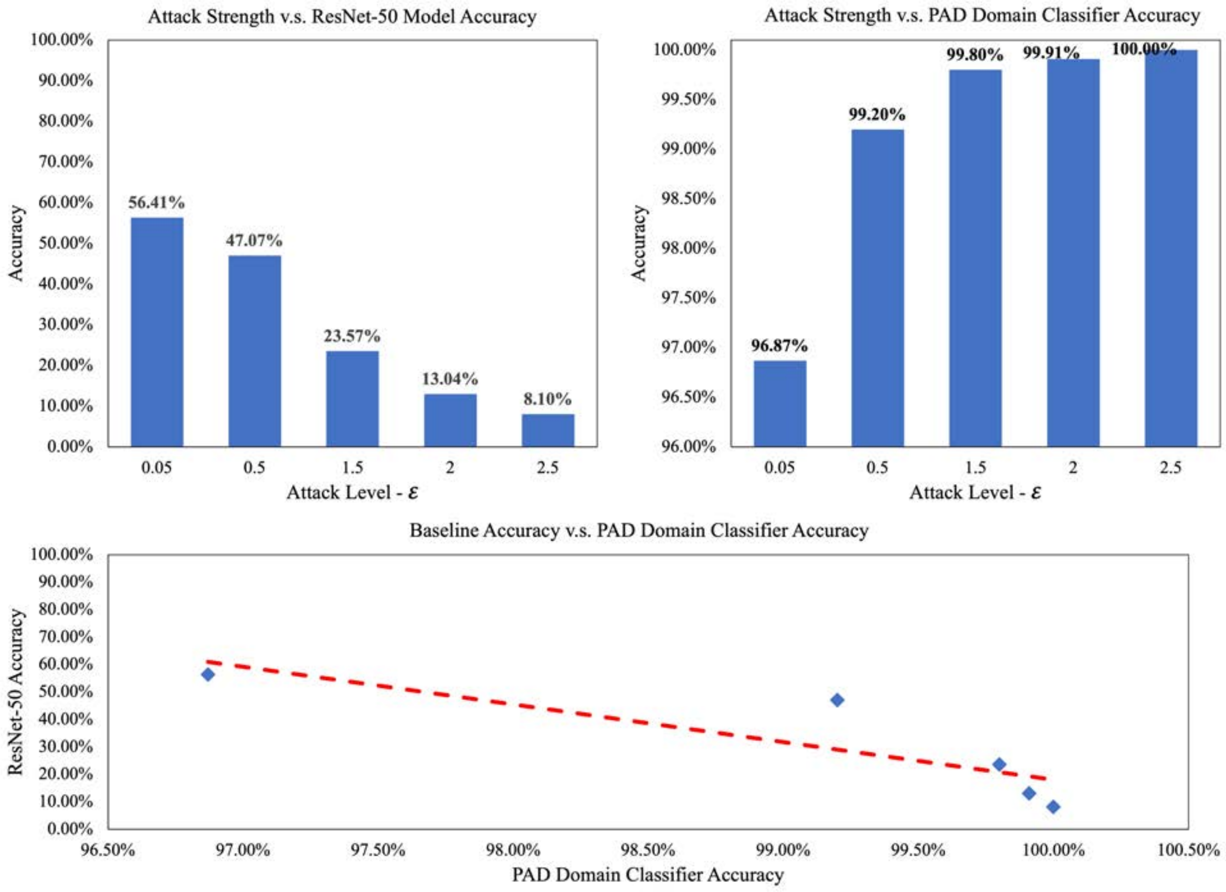
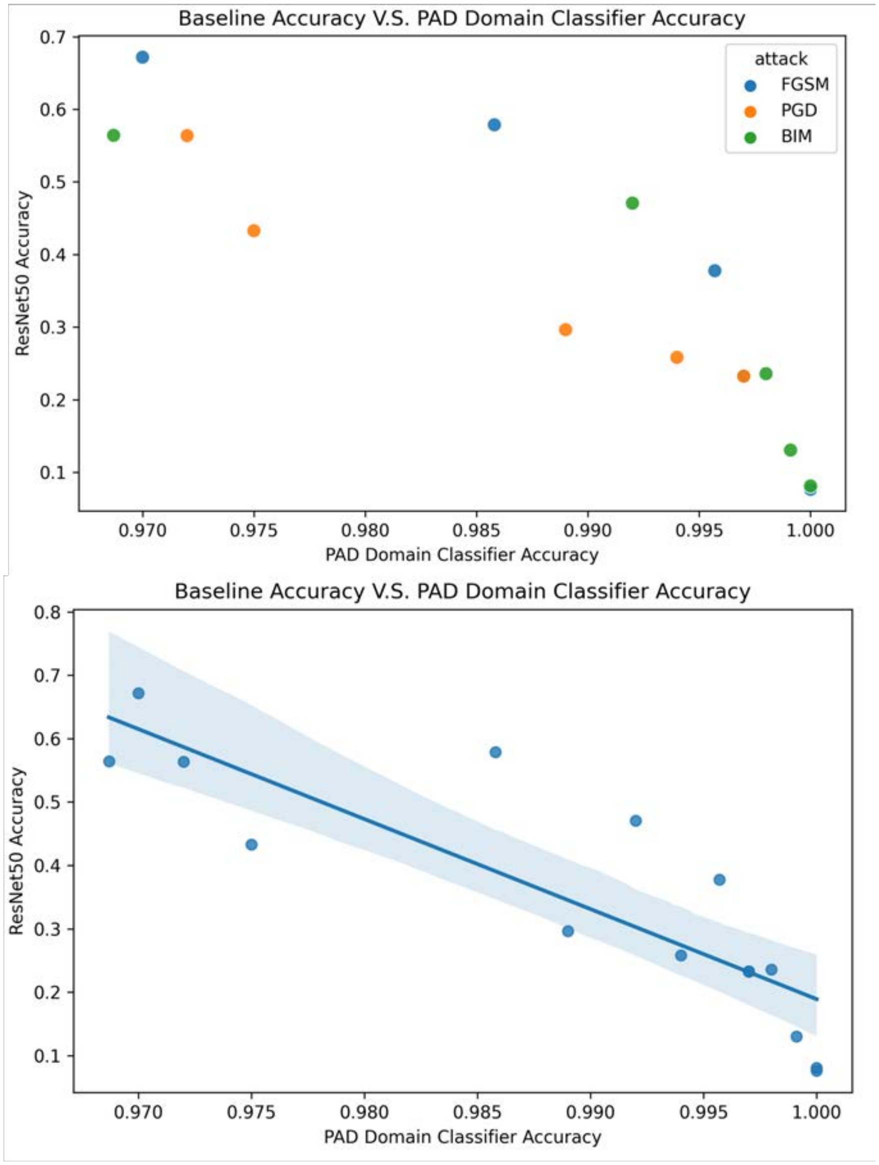


Figure 3.6: This plot indicates a positive relationship between attack strength and PAD measurement (a negative relationship between baseline model accuracy and PAD measurement). The change of PAD domain distributional shift measurements at each attack level validates that PGD adversarial attack causes distributional shift from the original images.

We summarized our calculation for PGD in Figure 3.6; The conclusion is similar to that of FGSM and BIM.

3.4 Summary of PAD Measurements for All Adversarial Attack Methods


Baseline Accuracy V.S. PAD Domain Classifier Accuracy

Baseline Accuracy V.S. PAD Domain Classifier Accuracy

Figure 3.7: Putting all three attacks together, we can see that PAD Domain Classifier Accuracy can be used as a proxy for measuring the strength of adversarial attacks. In other words, FGSM, PGD, and BIM attack image datasets by shifting their pixel distributions, and PAD can be used for quantifying the degree of the shift. The best-fitted line also indicates the same story.

Putting all three attacks, FGSM, BIM, and PGD, together, we can see that PAD domain classifier accuracy can be used as a proxy for measuring attack strength. Specifically, as PAD Domain Classifier accuracy increases, the baseline ResNet-50 model's accuracy decreases, which indicates a stronger attack strength. The best fitted line result (at the bottom of Figure 3.7) also indicates a negative relationship between baseline model accuracy and PAD Domain Classifier accuracy. From this study, we also notice that PAD measurement can be used as a standardized and universal measurements for all adversarial attacks. A standardized measurement allows side-by-side and universal comparisons between adversarial attack methods.

3.5 Training with Domain Adaptation Algorithms

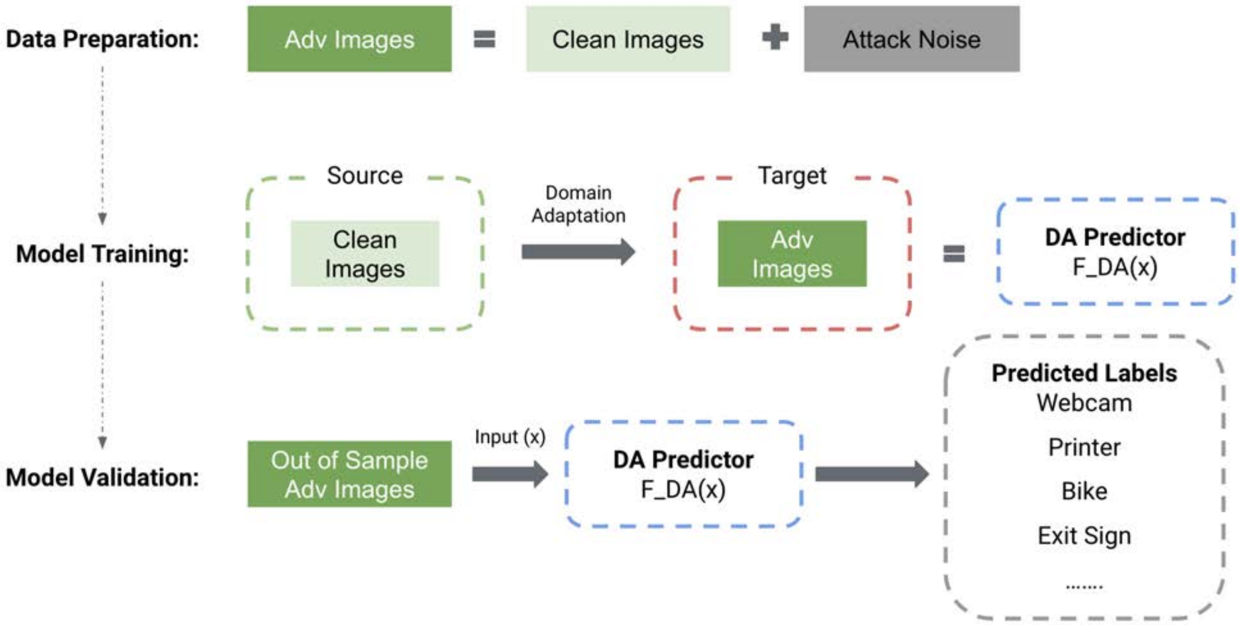


Figure 3.8: Training flow with Domain Adaptation algorithms. The procedure is: data preparation, model training, and model validation.

With adversarial data generated from FGSM, BIM, and PGD, we finished the data preparation step in the training flow chart (Figure 3.8). Then, we define clean images as the *Source Domain*, and adversarial images as *Target Domain*. For the Model Training sept, we train ResNet-50 model with DANN, JAN, CDAN as Domain Adaptation algorithms to create a model for each dataset. In the next chapter, we will analyze and summarized the results obtained from the last step of the training flow - model validation.

Chapter 4

Results

4.1 Overview

This chapter provides results generated from the experiment steps described in Chapter 3. As with the previous sections, we organize our results by attack methods to enable a side-by-side comparison between the accuracy of the Baseline ResNet-50 model and the Domain Adaptation models.

4.2 Domain Adaptation Algorithm Performance on FGSM Attacked Dataset

Table 4.1: Domain Adaptation Algorithms Accuracy on FGSM Attacked Dataset

	Clean	FGSM Attack Strength					Average Accuracy
		$\epsilon = 0.005$	$\epsilon = 0.1$	$\epsilon = 0.4$	$\epsilon = 0.8$	$\epsilon = 1.6$	
Baseline							
ResNet-50	96.57%	67.18%	57.56%	37.76%	23.30%	7.64%	38.69%
DANN	96.95%	91.39%	87.68%	81.96%	78.27%	49.16%	**77.69%**
JAN	96.98%	85.27%	78.31%	68.51%	60.27%	24.19%	63.31%
CDAN	97.01%	91.46%	87.01%	80.68%	72.69%	42.53%	74.87%

Note: Average Accuracy is the average of accuracy across 5 levels of attack strengths.

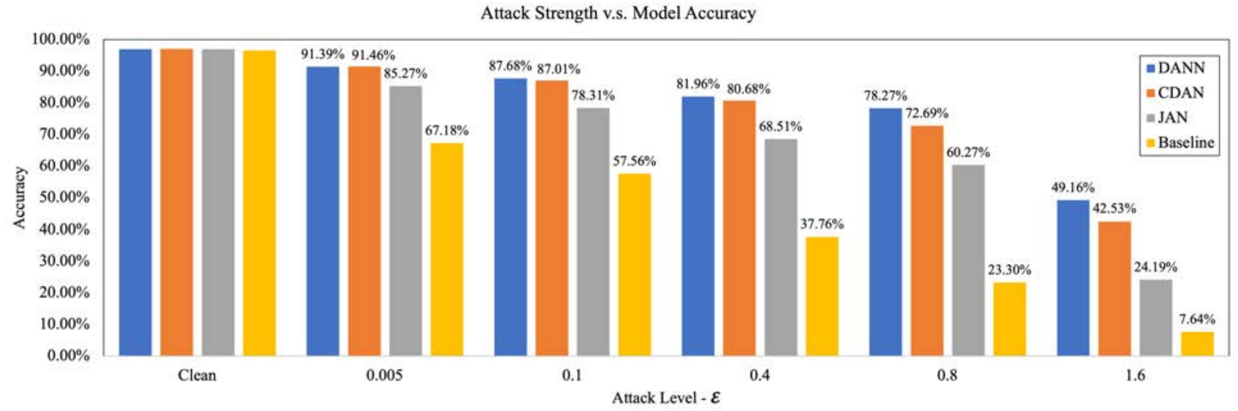


Figure 4.1: This plot provides a visualization of the models' accuracy across different levels of attack. While the baseline model suffers from FGSM attack, domain adaptation algorithms consistently perform better than the baseline. Within the domain adaptation models, DANN and CDAN consistently perform better than JAN.

4.3 Domain Adaptation Algorithm Performance on BIM Attacked Dataset

Table 4.2: Domain Adaptation Algorithms Accuracy on BIM Attacked Dataset

	Clean	BIM Attack Strength					Average Accuracy
		$\epsilon = 0.05$	$\epsilon = 0.5$	$\epsilon = 2$	$\epsilon = 3$	$\epsilon = 4$	
Baseline							
ResNet-50	96.57%	56.37%	43.72%	29.63%	25.82%	23.23%	35.75%
DANN	96.95%	88.04%	84.58%	77.37%	74.04%	73.03%	**79.41%**
JAN	96.98%	77.81%	68.74%	53.78%	50.20%	47.30%	59.56%
CDAN	97.01%	87.42%	83.06%	74.52%	70.23%	67.66%	76.58%

Note: Average Accuracy is the average of accuracy across 5 levels of attack strengths.

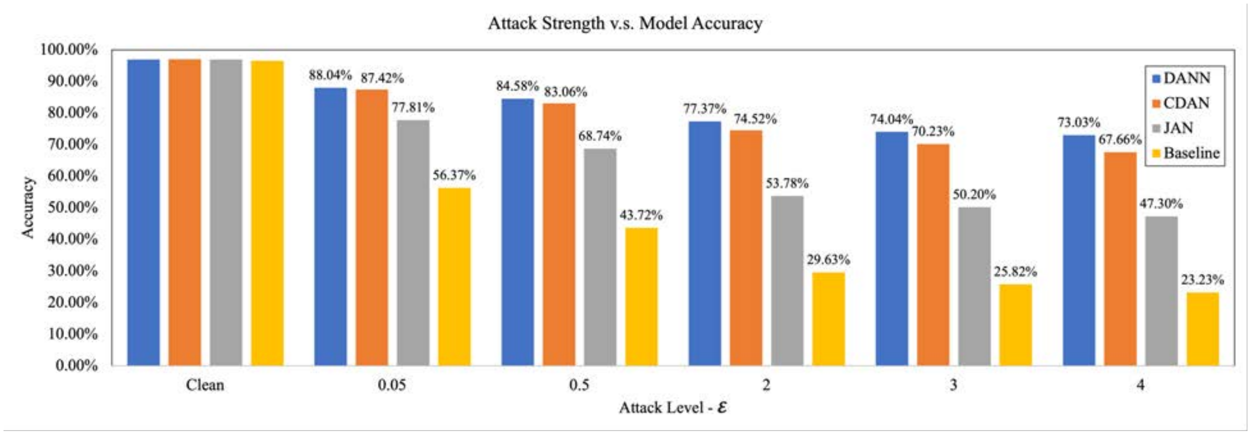


Figure 4.2: This plot provides a visualization on models' accuracy across different levels of attack. While the baseline model suffers from BIM attack, domain adaptation algorithms consistently perform better than the baseline. Within domain adaptation models, DANN and CDAN consistently perform better than JAN.

4.4 Domain Adaptation Algorithm Performance on PGD Attacked Dataset

Table 4.3: Domain Adaptation Algorithms Accuracy on PGD Attacked Dataset

	Clean	PGD Attack Strength					Average Accuracy
		$\epsilon = 0.05$	$\epsilon = 0.5$	$\epsilon = 1.5$	$\epsilon = 2$	$\epsilon = 2.5$	
Baseline							
ResNet-50	96.57%	56.41%	47.07%	23.57%	13.04%	8.10%	29.64%
DANN	96.95%	89.08%	88.57%	85.68%	77.69%	69.57%	**82.12%**
JAN	96.98%	78.77%	75.63%	67.78%	54.72%	42.09%	63.80%
CDAN	97.01%	88.69%	88.50%	83.43%	74.52%	66.15%	80.26%

Note: Average Accuracy is the average of accuracy across 5 levels of attack strengths.

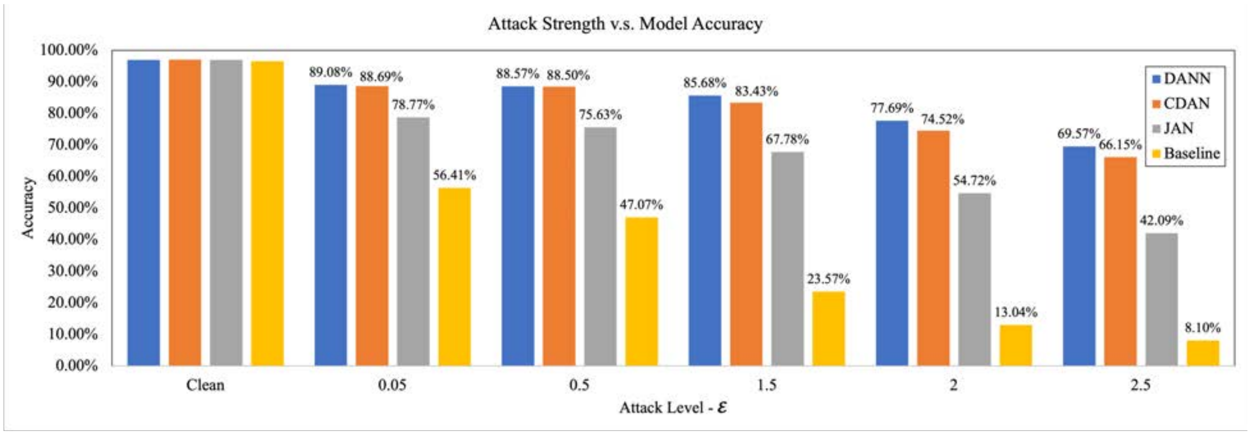


Figure 4.3: This plot provides a visualization on models' accuracy across different levels of attack. While the baseline model suffers from PGD attack, domain adaptation algorithms consistently perform better than the baseline. Within domain adaptation models, DANN and CDAN consistently perform better than JAN.

4.5 Summary and Observations

Table 4.1, 4.2, and 4.3 provide experimental results on defending adversarial attacks with Domain Adaptation (DA) algorithms. The first column of all three tables indicates that ResNet-50 trained with DA algorithms achieves the same accuracy as the regularly trained ResNet-50 baseline model on clean images. The same accuracy on clean images indicates that we are making a fair comparison across attacks because all models are equally good to start with. In terms of the overall pattern, we discover that DA algorithms consistently produce higher accuracy compared to the baseline ResNet-50 models across all attacks and all attack strengths. These results indicate that DA algorithms are relatively more robust to image adversarial attacks compared to the baseline model. We observe that while DA algorithms also have drops in accuracy as attack strength increases, the rate of decrease in accuracy is much lower than that of the baseline ResNet-50 model. In other words, DA algorithms are effective tools for defending adversarial attacks.

If we look closer into the accuracy trend within the DA algorithms (Figure 4.1, 4.2, 4.3), DANN performs the best compared to CDAN and JAN across all attacks. CDAN performs slightly worse than DANN but is still able to defend adversarial attacks effectively. Lastly, while JAN shows some ability in defending adversarial attacks, its performance is worse than those of DANN and CDAN. This observation inspires us to discover the theoretical explanation of how DA algorithms defend adversarial attacks. We provide detailed discussion and explanations of the results in Chapter 5.

Chapter 5

Analysis and Discussions

5.1 Overview

Results from chapter 4 indicate that simple DNN model trained with Domain Adaptation (DA) algorithms is effective in defending against image adversarial attacks. In this chapter, we will uncover the reason behind the advantages of DA algorithms through in-depth analysis and intuition discussions.

5.2 Domain Shift and Model Improvement

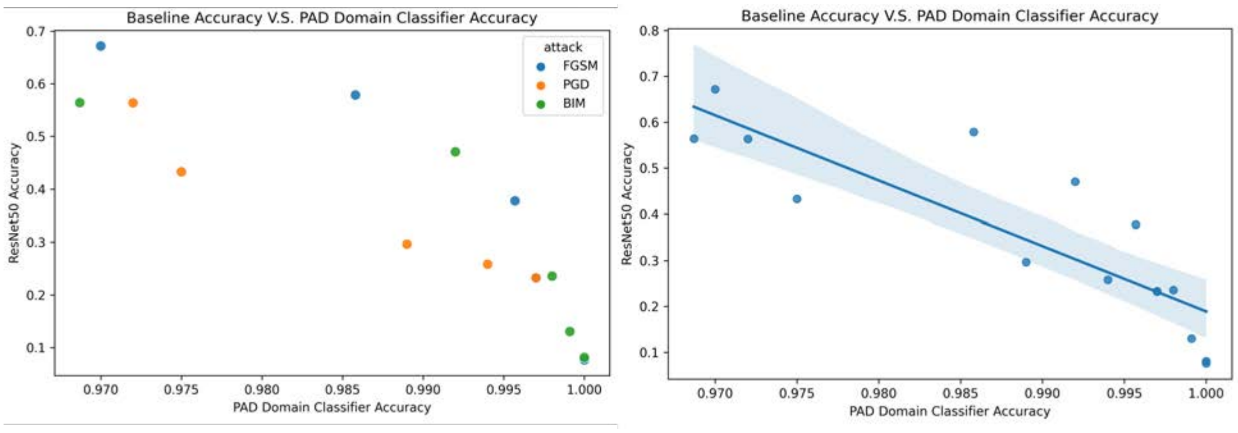


Figure 5.1: Proxy A-Distance is an effective method for detecting distributional shift.

As a reminder, the motivation behind using DA algorithms for defending against image adversarial attacks is that we consider such attacks to be a form of domain shift. In Chapter 3 and Figure 5.1, we discovered that Proxy A-Distance is an effective way to detect distributional shift of classical gradient-based image adversarial attacks (FGSM, PGD, BIM). Therefore, a natural next step in the analysis is to discover the relationship between the degree of domain shift and the improvements DA models have compared to the baseline model.

Define Model Improvements

We define model improvements as:

$$\textit{Improvement} = \textit{Domain Adaptation Accuracy} - \textit{Baseline Model Accuracy} \tag{5.1}$$

Taking a result table as an example (Table 5.1), we compute the improvement of DANN model compare to the baseline ResNet-50 model on in the last row. The larger the value, the better performance of a model under adversarial attacks. From the example in Table 5.1, we can see that DANN improves the baseline model acccuracy across all attack levels.

Table 5.1: Domain Adaptation Algorithms Accuracy and Improvement on PGD Attacked Dataset

	Clean	PGD Attack Strength					Average Accuracy
		$\epsilon = 0.05$	$\epsilon = 0.5$	$\epsilon = 1.5$	$\epsilon = 2$	$\epsilon = 2.5$	
Baseline ResNet-50	96.57%	56.41%	47.07%	23.57%	13.04%	8.10%	29.64%
DANN	96.95%	89.08%	88.57%	85.68%	77.69%	69.57%	82.12%
Improvement	0.38%	**32.67%**	**41.50%**	**62.11%**	**64.65%**	**61.47%**	**52.48%**

Note: Average Accuracy is the average of accuracy across 5 levels of attack strengths.

Domain Adaptation Model Improvements V.S. PAD Domain Shift

Following the definition of model improvements, we computed improvements for all models across all attacks and plotted them on Figure 5.2. In general, we model improvement increases as the distributional shift from the clean image domain becomes more significant. In other words, although DA models can be impacted by image adversarial attacks (Figure 4.1, 4.2, 4.3), they are more robust than the baseline model. The defending effect of DA models becomes stronger as the distributional shift increases.

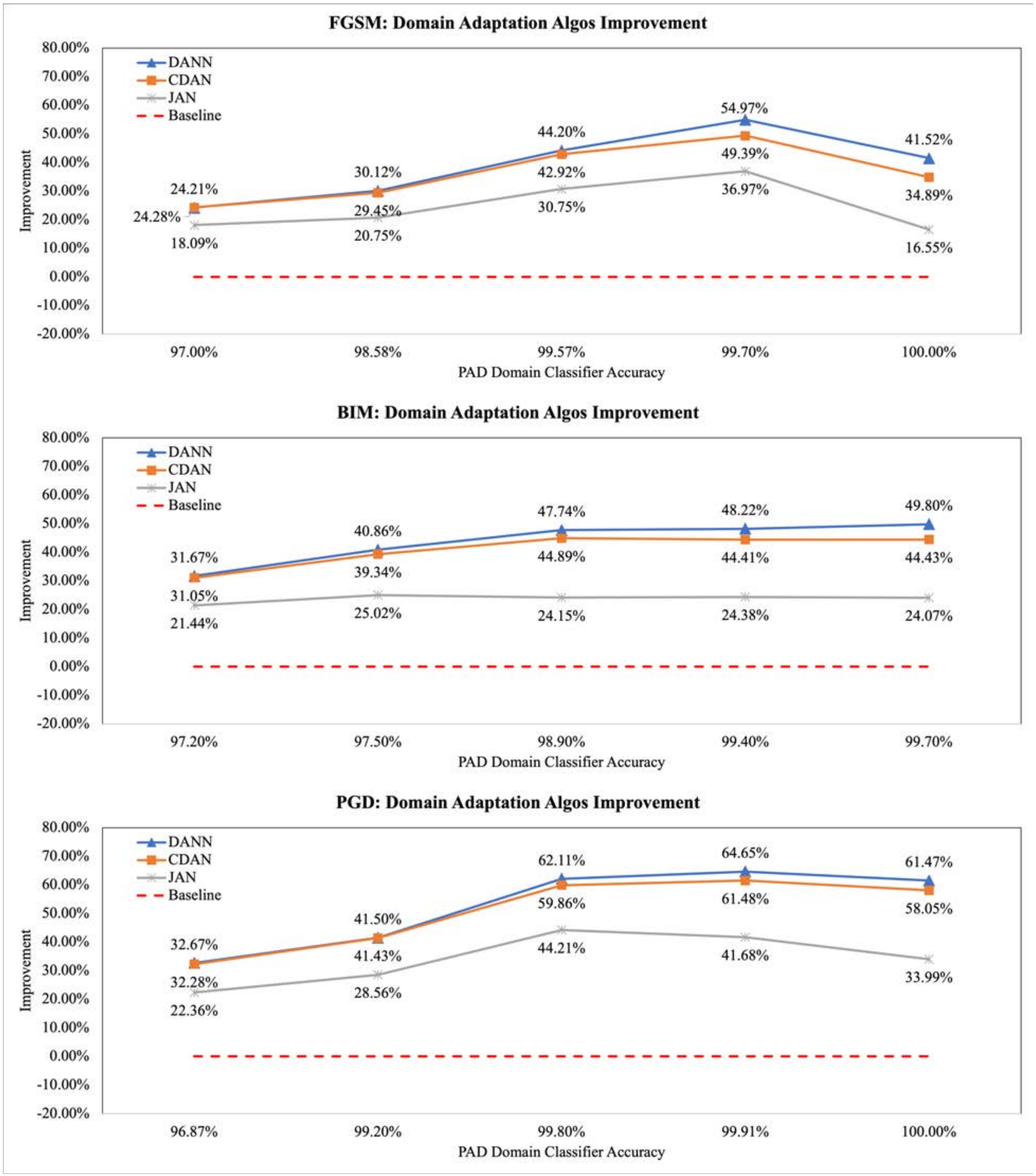


Figure 5.2: DA Model Improvements v.s. PAD Domain Accuracy for all attacks.

We also noticed that after the distributional shift becomes too strong, the improvement starts to drop. This can be inspected in the FGSM attack - the improvement begins to drop after a PAD measurement of 99.70%. We also found a similar pattern for BIM and PGD attacks. We think the reason behind the drop is that images become too noisy for any classifier to classify after a certain attack strength. For example, in Figures 3.1, 3.2, and 3.3, we can visually inspect that images can become so noisy that even humans cannot make confident predictions. Nevertheless, DA models are still extremely robust under adversarial attacks, even with high attack strength compared to the baseline model.

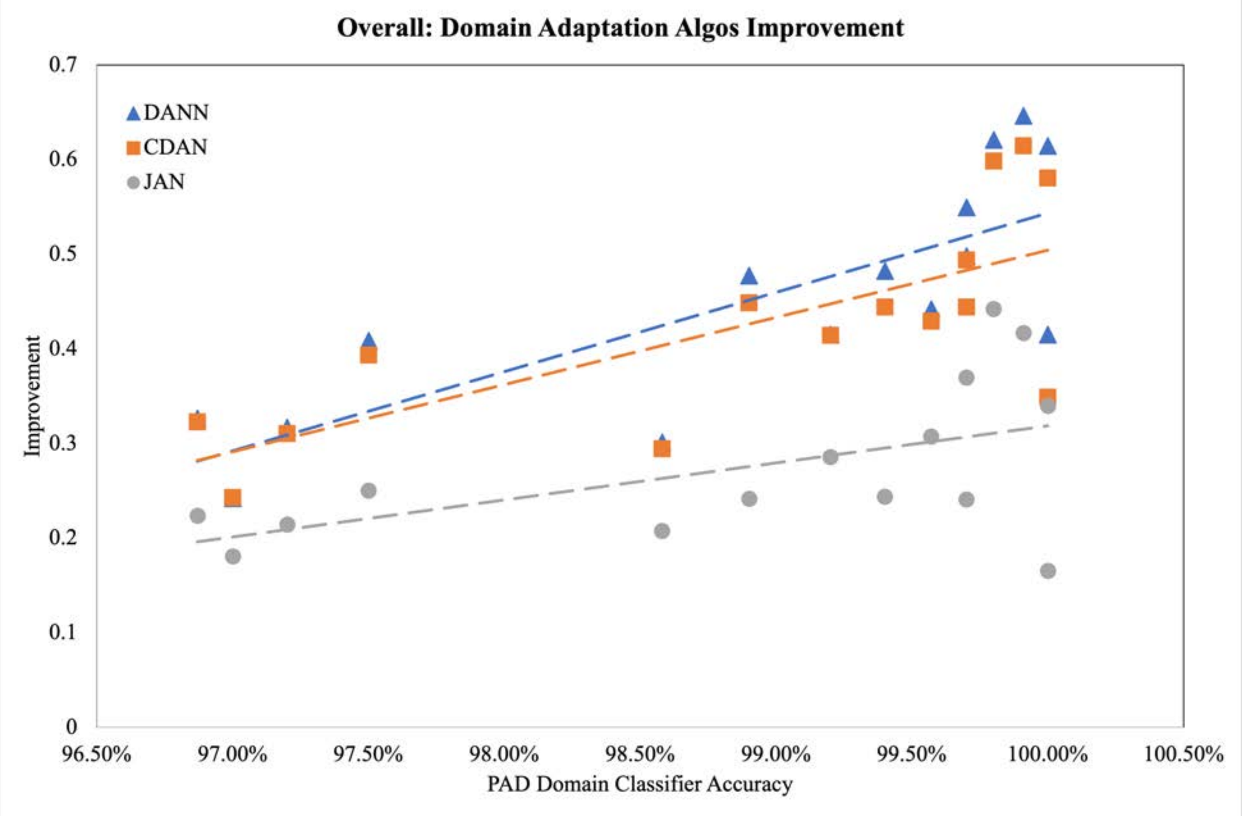


Figure 5.3: Overall DA Model Improvements v.s. PAD Domain Accuracy.

We constructed a scatter plot with best-fitted lines on the model level in Figure 5.3. In general, we see a positive relationship between domain distributional shift and DA model improvements on both the model level and overall level. Therefore, we think DA models are advantageous in defending against adversarial attacks that cause distributional shifts.

5.3 Reasoning Behind The Improvements

Domain Adaptation (DA) Algorithms show promising defenses against image adversarial attacks in our experiment results. Now, the question is, what makes DA models better than the baseline model?

The answer of the question is inspired by the performance discrepancies between DA models. Specifically, one can see in Figure 2.3 that JAN consistently performs worse than DANN and CDAN. While not too different from each other, DANN also tend to perform slightly better than CDAN in most of the attack cases. This is a counter-intuitive result because, based on the benchmark results (Table 2.1), CDAN should be the best model on classification tasks and JAN goes right after it with DANN being the worst [13][14]. This discrepancy in expectation made us wonder about the differences among the DA algorithms and how they impact the effectiveness in defending against adversarial attacks.

The Loss Functions

From the original papers, we realize that the biggest difference between the three DA algorithms is their loss functions [4][20][19]. To classify across domains, DA algorithms incorporate a domain loss component in addition to the regular classification loss. There are many ways to calculate domain loss from distribution differences, and we discover that DANN and CDAN use the same type of domain loss function (with different restrictions), while JAN used another type [5][19][20]. We provide mathematical definitions and reasoning of each Domain Adaptation methods in the following subsections.

DANN Loss Function

$$\mathcal{L}_{domain}\left(G_d\left(\mathbf{x}_i\right), d_i\right) = d_i \log \frac{1}{G_d\left(\mathbf{x}_i\right)} + \left(1 - d_i\right) \log \frac{1}{1 - G_d\left(\mathbf{x}_i\right)} \tag{5.2}$$

Where $G_d\left(\mathbf{x}_i\right)$ is a domain classifier (just like the Proxy A-Distance Domain Classifier) that takes in an image $\mathbf{x}_i$ and outputs $d_i = 1$ for *Target Domain* and $d_i = 0$ for *Source Domain* images. If the domain classifier makes the wrong prediction, the domain loss function will increase, which is how this function capture the distributional differences between two domains [5]. When the domain classifier can perfectly classify the domains of all images, the value of domain loss function will become zero.

CDAN Loss Function

$$\begin{aligned}\mathcal{L}_{domain}\left(G_d\left(\mathbf{x}_i\right), G_f\left(\mathbf{x}_i\right), d_i\right) = {} & d_i \log \frac{1}{G_d\left(\mathbf{T}(\mathbf{x}_i, G_f\left(\mathbf{x}_i\right))\right)} \\ & + \left(1 - d_i\right) \log \frac{1}{1 - G_d\left(\mathbf{T}(\mathbf{x}_i, G_f\left(\mathbf{x}_i\right))\right)}\end{aligned} \tag{5.3}$$

We define $\mathbf{T}$ as a multi-linear map or a randomized multi-linear map that convert two tensors to a single tensor. $G_f\left(\mathbf{x}_i\right)$ is an image classifier that predict the class label of image $\mathbf{x}_i$. Just like before, $G_d(\mathbf{x}_i)$ is a domain classifier (just like the Proxy A-Distance Domain Classifier) that takes in a tensor $\mathbf{T}(\mathbf{x}_i, G_f(\mathbf{x}_i))$ and outputs $d_i = 1$ for *Target Domain* and $d_i = 0$ for *Source Domain* images. This domain loss function is different from DANN as it adds a condition of image classifier into the loss function, which is a constraint in addition to the domain classifier [19].

JAN Loss Function

JAN loss function calculates domain loss via image Maximum Mean Discrepancy (MMD) as we discussed in chapter 2.4. Specifically, in the context of the original paper, the calculation is set up as follow [20]. We define *Source Domain* set as $\mathcal{D}_s$ with n_s labeled points. *Target Domain* set as $\mathcal{D}_t$ with n_t unlabeled points (because JAN is an unsupervised model). They are analogous to distribution P and Q from the definition of MMD in chapter 2.4. Then, in the JAN algorithm, the domain loss function uses activation generated in the *Source Domain* $\{(z_i^{s1}, ..., z_i^{s|\mathcal{L}|})\}_{i=1}^{n_s}$ and in the *Target Domain* $\{(z_i^{t1}, ..., z_i^{t|\mathcal{L}|})\}_{i=1}^{n_t}$. Lastly, we define a kernal function as k. Then, the domain loss function is defined as [13]:

$$\begin{aligned}\mathcal{L}_{domain}(\mathcal{D}_s, \mathcal{D}_t) = &\frac{1}{n_s^2}\sum_{i=1}^{n_s}\sum_{j=1}^{n_s}\prod_{l\in\mathcal{L}} k^l(z_i^{sl}, z_j^{sl}) \\ &+ \frac{1}{n_t^2}\sum_{i=1}^{n_t}\sum_{j=1}^{n_t}\prod_{l\in\mathcal{L}} k^l(z_i^{tl}, z_j^{tl}) \\ &- \frac{2}{n_s n_t}\sum_{i=1}^{n_s}\sum_{j=1}^{n_t}\prod_{l\in\mathcal{L}} k^l(z_i^{sl}, z_j^{tl})\end{aligned} \tag{5.4}$$

which computes an estimate of the squared distance between the empirical kernal mean embeddings generated from source and target domain DNN.

From the definitions of domain loss functions above, we can see that DANN and CDAN use domain classifiers, while JAN uses an MMD-based method. Based on observations in Chapter 2.4 and Figure 5.4, we confirm that the MMD-based method is unable to detect distributional shift caused by image adversarial attacks. Therefore, the discrepancies in performance among DANN, CDAN, and JAN are due to their differences in the type of domain loss function. Domain-classifier-based loss functions are able to detect adversarial attacks, which make DANN and CDAN more aware of the distributional shift caused by the attacks. On the other hand, the MMD-based method of JAN is unaware of the distributional shift and, therefore, unable to formulate a good loss objective against adversarial attacks.

In addition, CDAN seems to perform slightly worse than DANN across all kinds of attacks. If we look at Equations 5.2 and 5.3, they are pretty much the same, besides CDAN incorporating the output of an image classifier in the calculation. Based on the original paper, CDAN uses a conditional domain classifier to address the tradeoff between domain

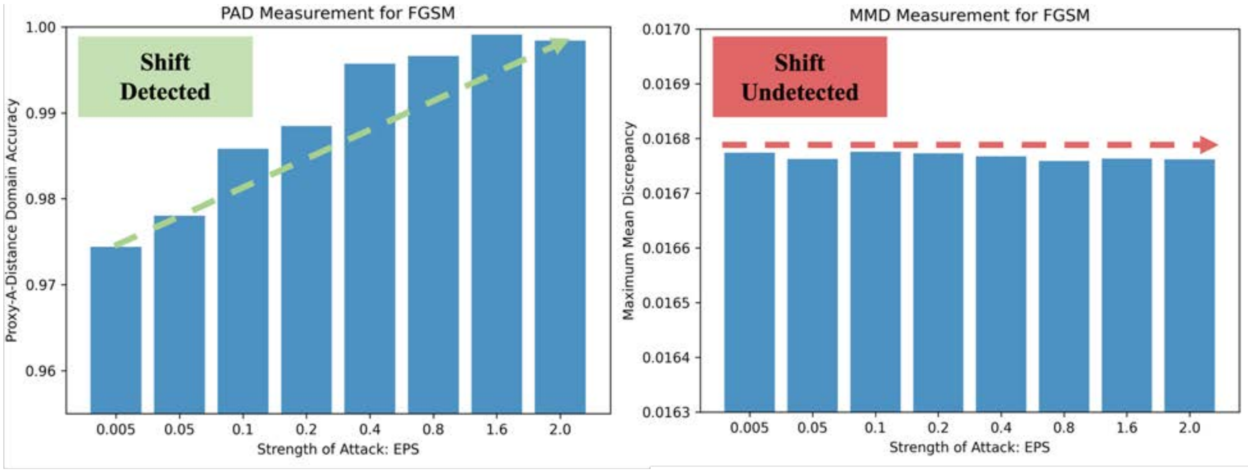


Figure 5.4: MMD cannot detect adversarial attacks like PAD

risk and classification risk. In the case of DANN, the domain loss function only considers the domain risk, which is exposed to less constraint. While CDAN can potentially achieve better performance in regular cross-domain classification tasks, the simplicity of DANN makes it a better model for defending against adversarial attacks. As we showed in Chapter 2.4, image adversarial attacks can be viewed as a form of domain distributional shift, which is the exact objective that the domain loss function of DANN is optimizing against. The additional conditions in CDAN increase the complexity of the optimization and, thus, undermines the direct benefit of the domain loss calculation.

Chapter 6

Extensions

6.1 Overview

We demonstrated the advantages and effectiveness of using Domain Adaptation (DA) Algorithms to defend against image adversarial attacks. All of our previous experiments were conducted using ResNet-50 as the baseline model and for domain adaptation training. Therefore, we extend the same study to ResNet-18 to avoid biases introduced by model size and to confirm that our method is able to achieve the same effect regardless of model complexity.

6.2 Domain Adaptation Algorithm Performance with ResNet-18

Tables 6.1 and Figure 6.1 indicate that DA models perform better than the baseline ResNet-18 model across all attack types and attack strengths. Similar to the case of ResNet-50, DA algorithms display the same effectiveness in defending against adversarial attacks with ResNet-18 DNN model. Furthermore, DANN algorithm remains to be the best performing algorithm among CDAN and JAN.

Compare to the performance with ResNet-50, Tables 6.1 and Figure 6.1 show a natural drop in accuracy due to the decrease in model size, which is expected for most of the DNN models. This extension study indicates that the defending effect of DA models is consistent across model complexities, which also shad lights on extending the same method to other types of state-of-the-art DNN models.

Table 6.1: Domain Adaptation Algorithms Accuracy on FGSM, BIM, and PGD Attacks with ResNet-18

	Clean	FGSM Attack Strength			Average Accuracy
		$\epsilon = 0.005$	$\epsilon = 0.8$	$\epsilon = 1.6$	
Baseline					
ResNet-18	94.64%	61.79%	20.59%	7.32%	29.90%
DANN	94.75%	87.54%	70.51%	37.96%	**65.34%**
JAN	93.98%	78.56%	53.09%	17.51%	49.72%
CDAN	94.01%	87.31%	68.56%	36.45%	64.10%
	Clean	BIM Attack Strength			Average Accuracy
		$\epsilon = 0.5$	$\epsilon = 2$	$\epsilon = 4$	
Baseline					
ResNet-18	94.64%	52.93%	45.81%	40.28%	46.34%
DANN	94.75%	82.92%	76.82%	74.66%	**78.13%**
JAN	93.98%	68.92%	60.41%	55.61%	61.65%
CDAN	94.01%	81.75%	76.18%	72.34%	76.76%
	Clean	PGD Attack Strength			Average Accuracy
		$\epsilon = 0.5$	$\epsilon = 1.5$	$\epsilon = 2.5$	
Baseline					
ResNet-18	94.64%	47.67%	25.80%	10.24%	27.90%
DANN	94.75%	84.65%	80.86%	65.53%	**77.01%**
JAN	93.98%	69.08%	56.60%	25.66%	50.45%
CDAN	94.01%	84.28%	79.21%	62.20%	75.23%

Note: Average Accuracy is the average of accuracy across 3 levels of attack strengths (ResNet-18).

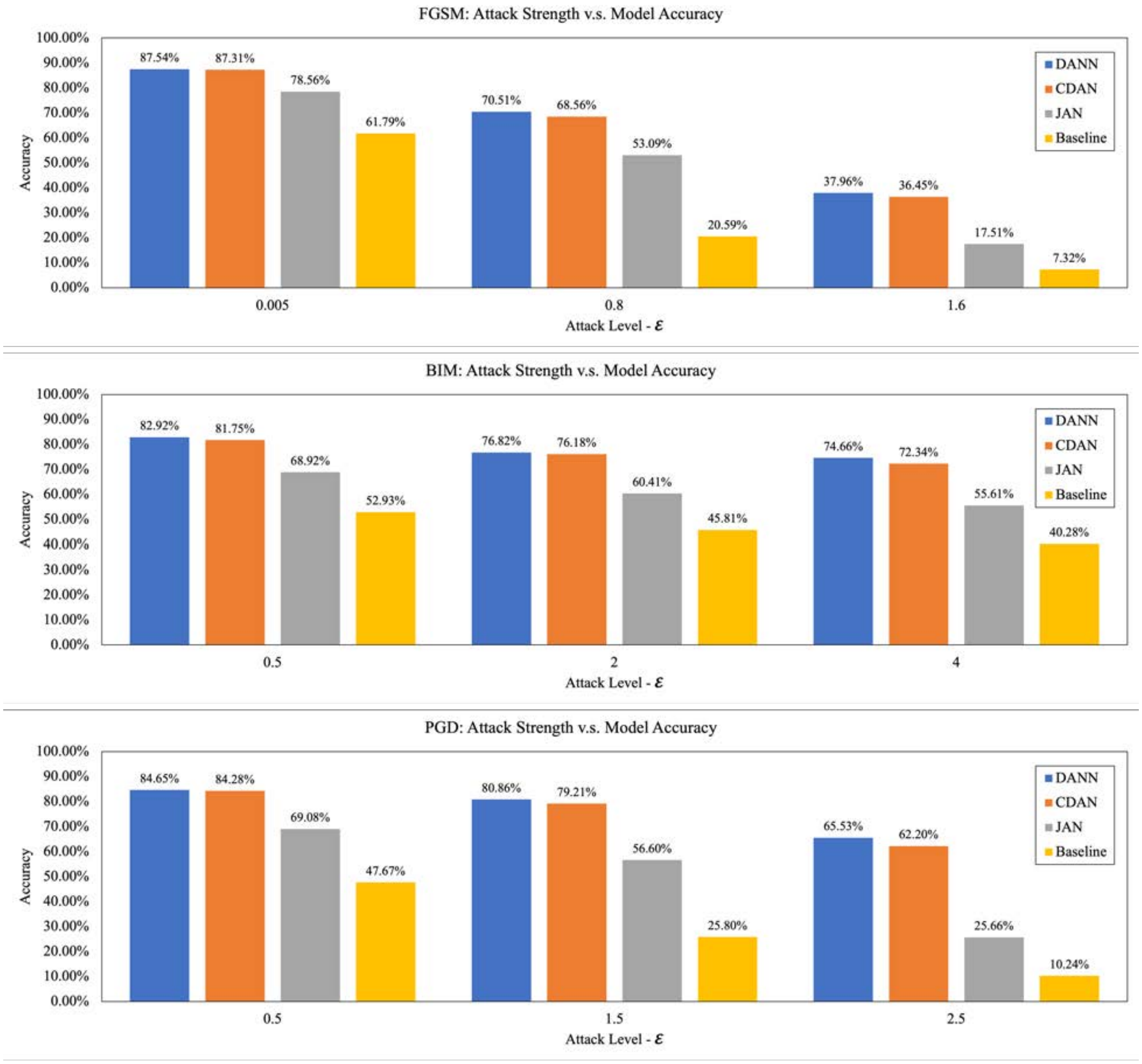


Figure 6.1: Performance summary of Domain Adaptations with ResNet-18 on all attacks and strengths

Chapter 7

Conclusion

7.1 Summary of Findings and Contributions

Domain Adaptation Defense (DAD) In this preliminary study, we formatted the image adversarial defense task into a cross-domain classification task with Domain Adaptation Algorithms. From the experiment results, we discovered that DA algorithms, such as DANN and CDAN, are effective methods to defend against classical gradient-based image adversarial attacks. We name this approach Domain Adaptation Defense (DAD). The key inspiration is to connect the distributional shifts caused by adversarial attacks with the advantages of the domain loss functions of DA algorithms. Specifically, by comparing the training processes across DA algorithms, we discovered that domain-classification-based loss functions can detect and quantify distributional shifts of adversarial attacks and thus enable DA algorithms to defend against attacks. DAD provides efficient and attack-function-agnostic defense, which is more practical in real-life applications. Lastly, we extended the same study on smaller DNN models and obtained the same conclusion. This indicates that DAD is applicable to DNN models with various complexities.

Distributional Shifts of Adversarial Attacks We designed procedures and conducted experiments to measure the distributional shifts of well-known and effective adversarial attacks. As a result, we confirmed that an adversarial attack is also a form of image distributional shift through Proxy A-Distance domain accuracy measurements. Besides demonstrating the effectiveness of Proxy A-Distance measurement, we verified that Maximum Mean Discrepancy (MMD) is unaware of distributional differences caused by adversarial attacks.

Proposed New Ideas to Apply Domain Adaptation In this report, we carefully outlined the experiment setup and the logic behind each measurement. As far as we know, we are the first experimental design to test the defense effect of DA algorithms against image adversarial attacks. With the promising results, we believe that our experiment set a new direction in the joint research area between adversarial attacks and DA algorithms, which will inspire new model architectures and distributional studies on adversarial attacks.

7.2 Future Directions

New Domain Adaptation Architectures Our study sheds light on potential directions in defending adversarial attacks and studying the distributional discrepancies in adversarial examples. Preliminary results indicate that existing DA algorithms can achieve promising accuracy under adversarial attacks, inspiring new DA algorithms designed specifically for adversarial attacks.

Study of Domain Loss Functions The key inspiration for this study is to identify the role of domain loss functions in measuring the domain distributional differences of adversarial examples. One direction for creating new domain adaptation (DA) algorithms is to revise the loss functions, focusing on measuring the domain shifts caused by adversarial attacks. Future researchers can experiment with various domain loss functions to test their awareness of adversarial attacks, which will help the community to further understand the limitations of these functions. Our experimental process provides the necessary tools to conduct experiments for better domain loss functions.

Better Understanding of the Defense Effects An obvious extension of our study is to simply conduct more experiments with more attack methods and DA algorithms. Since this is one of the first studies on this topic, we select the most well-studied methods for robust experiments. Obtaining more results will enable better understandings of the mechanisms and limitations of DAD.

Bibliography

[1] Shai Ben-David et al. "Analysis of representations for domain adaptation". In: *Advances in neural information processing systems* 19 (2006).

[2] Nicholas Carlini and David Wagner. "Adversarial examples are not easily detected: Bypassing ten detection methods". In: *Proceedings of the 10th ACM workshop on artificial intelligence and security*. 2017, pp. 3–14.

[3] Abolfazl Farahani et al. *A Brief Review of Domain Adaptation*. 2020. arXiv: 2010.03978 [cs.LG].

[4] Yaroslav Ganin and Victor Lempitsky. *Unsupervised Domain Adaptation by Backpropagation*. 2015. arXiv: 1409.7495 [stat.ML].

[5] Yaroslav Ganin et al. *Domain-Adversarial Training of Neural Networks*. 2016. arXiv: 1505.07818 [stat.ML].

[6] Ruize Gao et al. "Maximum Mean Discrepancy is Aware of Adversarial Attacks". In: *CoRR* abs/2010.11415 (2020). arXiv: 2010.11415. URL: https://arxiv.org/abs/2010.11415.

[7] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. "Domain adaptation for large-scale sentiment classification: A deep learning approach". In: *Proceedings of the 28th international conference on machine learning (ICML-11)*. 2011, pp. 513–520.

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

[9] Arthur Gretton et al. "A kernel two-sample test". In: *The Journal of Machine Learning Research* 13.1 (2012), pp. 723–773.

[10] Arthur Gretton et al. "Optimal kernel choice for large-scale two-sample tests". In: *Advances in neural information processing systems* 25 (2012).

[11] Keji Han, Bin Xia, and Yun Li. "(AD)2: Adversarial domain adaptation to defense with adversarial perturbation removal". In: *Pattern Recognition* 122 (2022), p. 108303. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2021.108303. URL: https://www.sciencedirect.com/science/article/pii/S0031320321004830.

[12] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *CoRR* abs/1512.03385 (2015). arXiv: 1512.03385. URL: http://arxiv.org/abs/1512.03385.

[13] Junguang Jiang et al. *Transfer Learning library.* https://github.com/thuml/Transfer-Learning-Library. 2020.

[14] Junguang Jiang et al. *Transferability in Deep Learning: A Survey.* 2022. arXiv: 2201.05867 [cs.LG].

[15] Hoki Kim. “Torchattacks: A pytorch repository for adversarial attacks”. In: *arXiv preprint arXiv:2010.01950* (2020).

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira et al. Vol. 25. Curran Associates, Inc., 2012. URL: https://dl.acm.org/doi/pdf/10.1145/3065386.

[17] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. “Adversarial examples in the physical world”. In: *CoRR* abs/1607.02533 (2016). arXiv: 1607.02533. URL: http://arxiv.org/abs/1607.02533.

[18] Yao Li et al. “A Review of Adversarial Attack and Defense for Classification Methods”. In: *CoRR* abs/2111.09961 (2021). arXiv: 2111.09961. URL: https://arxiv.org/abs/2111.09961.

[19] Mingsheng Long et al. *Conditional Adversarial Domain Adaptation.* 2018. arXiv: 1705.10667 [cs.LG].

[20] Mingsheng Long et al. *Deep Transfer Learning with Joint Adaptation Networks.* 2017. arXiv: 1605.06636 [cs.LG].

[21] Aleksander Madry et al. *Towards Deep Learning Models Resistant to Adversarial Attacks.* 2019. arXiv: 1706.06083 [stat.ML].

[22] Hemanth Venkateswara et al. “Deep hashing network for unsupervised domain adaptation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 5018–5027.

[23] Eric Wong, Leslie Rice, and J. Zico Kolter. *Fast is better than free: Revisiting adversarial training.* 2020. arXiv: 2001.03994 [cs.LG].

[24] Han Xu et al. “Adversarial Attacks and Defenses in Images, Graphs and Text: A Review”. In: *CoRR* abs/1909.08072 (2019). arXiv: 1909.08072. URL: http://arxiv.org/abs/1909.08072.

[25] Jiahui Yu et al. “CoCa: Contrastive Captioners are Image-Text Foundation Models”. In: *ArXiv* abs/2205.01917 (2022).

[26] Yi Zeng et al. *A Data Augmentation-based Defense Method Against Adversarial Attacks in Neural Networks.* 2020. arXiv: 2007.15290 [cs.CR].

[27] Wen Zhang and Dongrui Wu. “Discriminative Joint Probability Maximum Mean Discrepancy (DJP-MMD) for Domain Adaptation”. In: *2020 International Joint Conference on Neural Networks (IJCNN)*. 2020, pp. 1–8. DOI: 10.1109/IJCNN48605.2020.9207365.