東京大学
THE UNIVERSITY OF TOKYO

博士論文
Understanding Generalization in Neural Networks for Robustness against Adversarial Attacks
(敵対的攻撃に対するニューラルネットワークの汎化性能の理解)

チャウダリー　シュボジト

Subhajit Chaudhury

Student ID: 48-187411

Supervisor: Toshihiko Yamasaki

Department of Information and Communication Engineering

Graduate School of Information Science and Technology

The University of Tokyo, Japan

This dissertation is submitted for the degree of

Doctor of Philosophy

December 2020

Understanding Generalization in Neural Networks for Robustness against Adversarial Attacks

Abstract

Understanding Generalization in Neural Networks for Robustness against Adversarial Attacks

by

Subhajit Chaudhury

Doctor of Philosophy in Department of Information and Communication Engineering

The University of Tokyo, Japan

Supervisor: Professor Toshihiko Yamasaki

Deep learning methods are widely used in numerous commercial applications to their state-of-the-art performance in numerous domains. These applications include facial recognition, object detection, natural language understanding of textual input, acoustic analysis of user voices, and so on. Deep learning systems are also slowly making its mark on the medical and automated driving industry. Therefore, these modern machine learning methods are poised to tackle security-critical applications where a small mistake can cause substantial damage. Therefore, it absolutely essential to ensure that these models are safe to use and cannot be attacked by a malicious intruder to cause undesirable effects. The field of adversarial attacks exactly studies such vulnerabilities in Deep Neural Networks (DNNs) and provides methods for defense against such malicious attacks.

This thesis studies vulnerability in DNNs from the aspect of generalization. While most previous methods study vulnerabilities during test-time attacks, we analyze DNN vulnerabilities by corrupting the training images. Specifically, we first propose a gradient-free method for finding training time attacks that reduce accuracy on clean test images. We also analyze the performance of various regularization methods and loss objectives showing that vanilla cross-entropy loss is vulnerable against our attack. Additionally, using our proposed attack, we show that Stochastic Gradient Descent (SGD) based optimizers are robust against training time noise compared to adaptive optimizers (such as Adam) which are a popular choice of DNN optimizers. Furthermore, we show that such training time attacks can jeopardize security-critical applications like medical image analysis by significantly reducing model accuracy on test images. Finally, we show that models learned in the frequency domain result in better model robustness against adversarial and spatial transformation attacks, due to frequency disentanglement between adversarial nuisance and semantic features. We believe this thesis presents useful tips and tricks with experimental evidence towards DNN training that will help researchers and practitioners in training robust models under adversarial noise.

Acknowledgments

This thesis is a combination of the research work that was conducted in the last three years. These last three years constituted both a challenging and enjoyable journey that inculcated qualities in me that would help me shape my future career and life. Firstly, it taught me that only having a brilliant idea or being intelligent is not enough to succeed in life. Being patient, persevering against initial failures like paper rejections, and then standing back up against all odds is very important for any new undertaking in life. Now after having gone through the Ph.D. journey, I can confidently say that I can tackle most problems that life throws at me and obtain a fruitful solution.

Although this was a personal journey, it would not have been possible without the immense support of numerous people. First, I would like to express my sincere gratitude to my supervisor Professor Toshihiko Yamasaki for his invaluable guidance and academic support. He has taught me the most valuable lesson that failures such as paper rejections are not a reflection of personal inabilities but rather it might just not be the right time for success for various external factors like competition, random assignment of reviewers, and other such factors. What we should do is give our best performance every time without compromise and let the universe decide the outcome of that work. This is practical and valuable advice that has led me to success in many cases even outside of my Ph.D. life. Therefore, I learned from him how to stay positive in the face of adversity and stay confident even in the hardest times. After every meeting with him, I could always have a positive and hopeful mindset about my research direction that gave me great motivation to conduct research and submit to top-quality conferences. Additionally, I would also like to thank him for providing the financial support that was of immense help to me for covering my tuition fees. I would also like to thank Prof. Matsui and Prof. Aizawa for asking relevant questions during the lab presentations that helped me shape my research direction and paper contents when submitting to conferences.

Of course, my peers gave me immense support during my Ph.D. life. I would like to thank Hiya for making me part of a super cool paper on using computer vision techniques on martian surface images with collaboration from NASA and JAXA. Thanks for helping with all the administrative stuff that I could not comprehend due to my limited Japanese ability. Also, for my sincerest gratitude for considering my uncountable requests for helping me finish our papers on time with good aesthetics. I would also like to thank Sourav for making me part of the super impactful research on using machine learning on skin images. Although I would be lying if I said I enjoyed looking at the skin disease images. However, I strongly believe this research has the potential to help many people to diagnose skin ailments at affordable rates, especially for people from the poor section of society. Additionally, I would like to thank Naoto Inoue for discussions on my research topics. I also got motivated to conduct good research by reading your CVPR and AAAI papers. I would also like to thank post-doctoral researchers, Xueting Wang, and Hiroshi Kera for great discussions on my research topics that helped in shaping my future directions. Finally, I would like to thank all other members in the lab who gave me comments for my Rinko and Pre-defense presentation.

Furthermore, I would like to thank all my friends outside Ph.D. life for making my life during the grueling research struggles enjoyable by arranging drinking parties and house parties. I am also grateful to my management and peers at IBM Research who have supported my Ph.D. at the University of Tokyo. I would also like to thank my sister Puja for supporting me during my Ph.D. struggles. Finally and most importantly, I would like to thanks my parents, without whom I would exist on this planet. They were extremely supportive in my Ph.D. endeavors and I would like to thank them for all my success so far leading up to my PhD.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Due to the availability of large amounts of data and strong computational power, Deep Neural Networks (DNNs) are poised to capture large segments of the commercial markets. Recent works in Convolutional Neural Networks (CNNs) [42, 46, 85, 93] have showcased their immense academic and commercial contribution due to their notable empirical success in various application areas. Similarly, transformers based models [23, 82, 104] have shown state-of-the-art performance in natural language processing applications. With such wide-spread commercial applications, it becomes paramount that we ensure the safety and security of such applications. Recent works in adversarial attacks [6, 16, 63, 98] expose such vulnerabilities in neural networks which demands further study to ensure the safe useability of such methods.

In this thesis, we propose a special kind of attack that exposes the vulnerability of CNNs to training time attacks. Specifically, we outline an evolutionary strategy based attack that corrupts automatically selected pixels in the training images that maximally reduces accuracy on clean test images. Using this attack method, we analyze the performance of neural network properties related to generalization and prevention of overfitting in the presence of regularization methods. We also analyze various loss functions and show that vanilla cross-entropy loss which is widely used in various deep learning classification tasks are not robust to training noise and hence leads to poor performance in the presence of training noise. We propose an improved training loss that considers the mutual information between the learned features and the labels, which shows improved performance in the presence of training noise. Furthermore, we use our method to benchmark popular neural network optimization methods and show that SGD based methods are more robust compared to adaptive optimization methods in the presence of training noise. We further show that our proposed attack methods can also attack security-critical applications like medical image classification etc, thus showing the impact of such training time noise attacks. Finally, we perform a frequency-domain analysis of the adversarially attacked images and show that the CNNs trained on frequency domain input shows better robustness to adversarial attacks. We hypothesize that adversarial noise and semantically useful features occupy different frequency range that can be disentangled by the CNN learned in the frequency domain thus enabling best adversarial robustness.

Before going in the details of our proposed method, we will present for background on previous adversarial attack methods that will motivate the need for finding security measures to improve neural network performance to such attack algorithms. In the following section, we outline various such attack methods both 'evasive' and 'poisoning' attack methods and also described some adversarial defense methods.

1.1 Adversarial Attacks

Adversarial attacks be broadly classified into two categories: (i) evasive attacks, where, the model is trained on clean images and adversarial attacks is performed on the query image, (ii) poisoning attacks, where, the training images contain some noisy data on which when the model is trained, it results in back door attacks for some special images in the clean test set. Below describe some methods for both kind on attacks:

1.1.1 Evasive Attacks

Consider a classification model $f(x;\theta)$ parameterized by θ, where $(\boldsymbol{x}, y)$ In evasive attacks, there are two kinds of attacks, white-box attacks and black-box attacks. In white-box attacks, attacker has full knowledge of the model parameters and also the data label, $(\boldsymbol{x}, y)$. In black-box models, the attacker does not have full access to the model parameters, but has access to the training data. First we describe some white box attacks and then we described few black box evasive attacks.

Fast Gradient Sign Method

This is a single step attack method [35] and hence a very fast white-box method to find adversarial examples. Starting from the original image $\boldsymbol{x}$ with label y and the loss function $\mathcal{L}(\boldsymbol{x}, y, \theta)$. The adversarial non-targeted attack is computed as

$$\boldsymbol{x}' = \boldsymbol{x} + \epsilon \nabla_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, y, \theta), \tag{1.1}$$

where the $\nabla_{\boldsymbol{x}} \mathcal{L}$ computes the gradient of the loss function (typically cross-entropy loss) around the image value. Since it just involves a single step, it is quite fast however not very effective.

Projected Gradient Descent

This method computed adversarial attacks using multiple iterative steps to compute the best adversarial noise that maximizes the loss function causing misclassification on test image. The formula to compute the attack is given as

$$\delta := \mathcal{P}_{\Delta}[\delta + \nabla_{\delta} \mathcal{L}(\boldsymbol{x} + \delta, y, \theta)]. \tag{1.2}$$

Here the $\mathcal{P}_{\Delta}$ is the projection operator on the feasible region defined by Δ which is defined by distance norm such as ℓ_0, ℓ_1, ℓ_2 or ℓ_∞. Since this is an iterative method, it is usually stronger than single step attacks.

Universal Adversarial attack

While the previous methods are limited to single attack pattern for each image, the universal adversarial attack [66] finds a single attack δ such that it causes misclassification for all images in the test set. This is formulated as

$$\mathbb{P}_{\boldsymbol{x},y\sim\mathcal{D}}[f(\boldsymbol{x}+\delta)\neq f(\boldsymbol{x})] \quad s.t.||\delta||_p<\epsilon, \tag{1.3}$$

where the attack δ is designed to attack all images sampled from the image distribution $\mathcal{D}$.

One Pixel Attacks

In this attack [97], just one pixel is superimposed on the test images and the corresponding test accuracy is computed. Therefore, this method puts a constraint on the ℓ_0 norm by limiting the number of pixel attacks.The best location of the pixel attack is determined by gradient-free evolutionary approach. This method shows that even a single pixel attack can cause significant drop in accuracy thus showing serious vulnerabilities in neural network learning.

1.1.2 Poisoning Attacks

Poison attack methods alter few training images instead of attacking the query images. Here we outline some methods for poison attacks.

Biggio et al. attack

This method [12] is one of the first poisoning methods that add some noisy images in the training set such an Support Vector Machines (SVM) trained on such data results in high loss on the validation set. This is performed by finding which training samples contribute to the decision function of the SVM. This idea was extended to the neural network domain to find poison images as well by later models.

Influence Functions

This method [55] uses interpretability in deep neural networks based on the theory of influence functions to find how the DNN predictions change by changing the training images. They use this method to find selective training images that can change the label of targeted images in the test set also known as back door attacks.

Poison Frogs

The PoisonFrogs [91] work considers a feature extractor $Z(\boldsymbol{x})$, that extracts a semantic features from raw images. Given a target image $\boldsymbol{x}_t$ and target class y_t in the test set, this method add another image $\boldsymbol{x}'$ in the training set that is very close in the feature space such that $||Z((\boldsymbol{x}_t) - \boldsymbol{x}'||$ is minimized. The initial value for $\boldsymbol{x}'$ is taken from a base image that is changed to obtain the final value. Since the model is trained on $\boldsymbol{x}'$ that is very close to another sample $\boldsymbol{x}_t$, the model will associate the target label thus causing misclassification.

1.2 Adversarial Defenses

Defensive strategies towards adversarial attacks can be categorized into Gradient Masking and Robust Optimization. Here we explain some methods from these defensive strategies.

1.2.1 Defensive Distillation

Hinton et al. [44] introduces distillation mainly used for model compression and better generalization in neural networks. Papernot et al. [76] used the defensive distillation idea to train a DNN at higher temperature T from soft labels from another teacher network. During prediction, they used the setting, $T = 1$ that prevented gradient based attacks to successfully create adversarial samples.

1.2.2 Shattered Gradients

Various defensive methods use non-differentiable input transformations to stop gradient-based attacks from gradient propagation. Input transformation based methods like [38, 80] use methods like image cropping, resizing, randomly changing pixel order to defend against adversarial attacks. Thermometer encoding [15] discretize the input values from floating point to n-bit vectors and training is performed on that input.

1.2.3 Adversarial Training

The previous two approaches constituted the Gradient Masking approach for adversarial defense. The method of adversarial training falls under robust optimization kind of defense against adversarial attacks. In this method, for each mini-batch of data, we compute the corresponding adversarial samples and append them as the training set. Models trained using PGD [63] show better robustness compared to other methods like FGSM [35] used by Kurakin et al. [56].

1.3 Thesis Summary

This thesis outlines various methods for adversarial attacks in training images, explain various properties of DNNs using such attacks and proposes various techniques for adversarial defense. In Chapter 1, we outline the summary of this thesis and explain related works for adversarial attacks and defenses related to this work.

In Chapter 2, we propose a novel evolutionary strategy based algorithm, called *EvoShift*, for optimizing pixel attacks that are added to the training images. To obtain our training time attack, we solve a joint $\min\max$ optimization with the outer maximization designed to find the pixel noise and inner minimization designed to train the neural network on the noisy images. We impose a constraint on the optimization such that the cross-entropy (CE) loss on the noisy images is low, and the loss on the clean images is high. Such a formulation results in CNN trained on the noisy images to have a very high error on clean test images, thus exposing serious vulnerabilities in CNNs that are detrimental to robust learning. Interestingly, we find that optimization choice plays a vital role in generalization robustness. We show empirical evidence that SGD is resilient to our training time attacks, unlike adaptive optimization techniques (Adam). Although adaptive optimization methods are a popular choice for practitioners and researchers, we show that they can easily overfit in the presence of our training time attacks. We believe that this is an important finding for the machine learning research community. We also apply regularization methods to counteract our proposed adversarial training time attack and find that most well-known regularization methods are ineffective against our attack. We find that random-crop data augmentation is moderately effective for a few pixel attacks. As a defense against our attack, we propose a novel robust loss function for CNN classification training that is resilient against our training attacks using an information maximization framework. This result suggests that the traditional cross-entropy minimization framework for CNN training might cause non-robust feature learning, which might be mitigated by our proposed information-theoretic loss function. We introduce the concept of vulnerability in Generative Adversarial Networks (GANs) under proposed *EvoShift* attacks, causing poor image generation quality due to overfitting in the GAN discriminator.

In Chapter 3, we train DNN models on noisy training data using various optimizers and measure the performance of such models on clean test data, thus benchmarking how liable the optimizers are to overfit the training noise. We first construct a linearly separable two-class toy dataset upon which we superimpose a crafted noisy signal. Following that, we analytically show that adaptive gradient methods completely fail to learn any patterns from the data and do not generalize to the clean test set. On the other hand, SGD and its variants show 100% test accuracy on the test-set showing greater robustness against such spurious noise. Secondly, for higher-dimensional image datasets, we use a gradient-free noise optimization, based on [18] for finding optimal pixel perturbations that maximize the generalization gap between training and testing images. Convolutional Neural Networks (CNN) models are trained on such worst-case noisy images using various optimizers. These trained models are then evaluated on clean data to measure the generalization of optimization methods. Empirical studies on MNIST, CIFAR10, and SVHN dataset confirm our hypothesis that vanilla SGD and its variants are significantly more robust against such perturbations compared to adaptive gradient methods. Our analysis of the 2D loss surface reveals that SGD tends to find solutions around flatter loss regions, which might explain our empirical observations. Based on our benchmarking results, we recommend using SGD optimizers with learning rate tuning instead of adaptive gradient methods, especially when there exists some training noise or distribution shift between the training and validation/testing data.

In Chapter 4, we expose the vulnerability of deep learning models for analyzing medical images under *worst-case* few pixel perturbations on training images. We emulate practical scenarios where few dots (almost imperceptible to human eyes) have appeared on the medical image because of some device noise. Therefore, we show that if the model is trained using those single/few pixels perturbed images, the network learns absolutely nuisance features instead of useful semantic features and provides unexpectedly low test accuracy. We utilize the evolutionary strategy based few pixel perturbation algorithm from [19] to corrupt the training images that maximally ignores the task-relevant features (like shape and appearance) due to over-reliance on spurious distractor artifacts. We benchmark popular deep learning models on medical image datasets under such training time noise and show that even with single-pixel perturbations, deep models are susceptible to overfitting behavior similar to random classifiers. Informed by this analysis, we study input Covariate Shift Normalization (CSN), to reduce the effect of such spurious predictive features. Additionally, we analyze vanilla SGD and adaptive optimizers under such pixel perturbations in medical images and show that SGD is surprisingly more robust than adaptive methods (like ADAM) which is a default choice of optimization for most practitioners.

In Chapter 5, we show that adversarial features occupy a separate region in the frequency spectrum that can be disentangled from the regions occupied by semantically meaningful features in natural images. We use this concept to propose an adversarial defense against popular adversarial attacks. We empirically show that learning in the frequency domain can be used as a defense against adversarial images by a feed-forward operation of the frequency domain transformation of the input adversarial image through the frequency CNN. This method of defense outperforms previous input transformation based adversarial defense methods. Finally, we show that our method is robust against spatial transformation attacks such as rotations and translations, to which naturally trained CNNs show poor performance as shown by the work of [28].

Finally in Chapter 6, we provide conclusion from our experiments. This thesis provides empirical evidence regarding various properties of neural network training that we believe are of significance to the deep learning community. We hope this thesis will instigate future research on robustness of neural networks with respect to generalization and optimization techniques.

Chapter 2

Adversarial Training Time Attack against Discriminative and Generative Convolutional Models

In this chapter, we show that adversarial training time attack by a few pixel modifications can cause undesirable overfitting in neural networks for both discriminative and generative models. We propose an evolutionary algorithm to search for an optimal pixel attack using a novel cost function inspired by domain adaptation literature to design our training time attack. The proposed cost function explicitly maximizes the generalization gap and domain divergence between clean and corrupted images. Empirical evaluations demonstrate that our adversarial training attack can achieve significantly low test accuracy (with high train accuracy) on multiple datasets by just perturbing a single pixel in the training images. Even under the use of popular regularization techniques, we identify a significant performance drop compared to clean data training. Our attack is more successful than previous pixel-based training time attacks on state-of-the-art CNNs architectures, as evidenced by significantly lower testing accuracy. Interestingly, we find that the choice of optimization plays an essential role in robustness against our attack. We empirically observe that Stochastic Gradient Descent (SGD) is resilient to the proposed adversarial training attack, unlike adaptive optimization techniques, like the popular Adam optimizer. We identify that such vulnerabilities are caused due to over-reliance of cross-entropy loss on highly predictive features. Therefore, we propose a robust loss function that maximizes the mutual information between latent features and input images, in addition to optimizing cross-entropy loss. Finally, we show that the discriminator in Generative Adversarial Networks (GANs) can also be attacked by our proposed training time attack resulting in poor generative performance. Our method is one of the first works to design attacks for generative models.

2.1 Introduction

Convolutional Neural Networks (CNNs) [42,46,93] are a powerful class of models that learn hierarchical feature representations and have shown notable empirical success in various application areas. Typically, in an over-parametrized setting with a highly non-convex loss surface, classical learning theory [103] predicts that these deep neural network models should have a high out-of-sample error because the solution is likely to get stuck at a local minimum. Nonetheless, deep neural networks appear to generalize well, even in small data regimes. Numerous previous works have shown that current deep learning models are not robust against adversarial attacks [6,16,63,98]. However, due to these models' notable empirical success in various application areas such as computer vision [42, 85], natural language processing [23, 79, 104], and other real-world domains, deep learning is poised to lead us to the next industrial revolution. The increasing use of such machine learning models in security-critical applications and the unpredictable behavior of these models under tiny well-crafted perturbations demands a better understanding of neural network robustness to ensure safe, practical implementations.

Traditional methods in adversarial attacks [16, 21, 26, 35, 56, 63, 68, 75, 89, 98] fool trained neural networks using *adversarial* query images. These attacks add small perturbations to the query images resulting in the classification function to cross the true class's decision boundary causing incorrect classification. However, such perturbations are imperceptible by humans, making them difficult to detect. Such

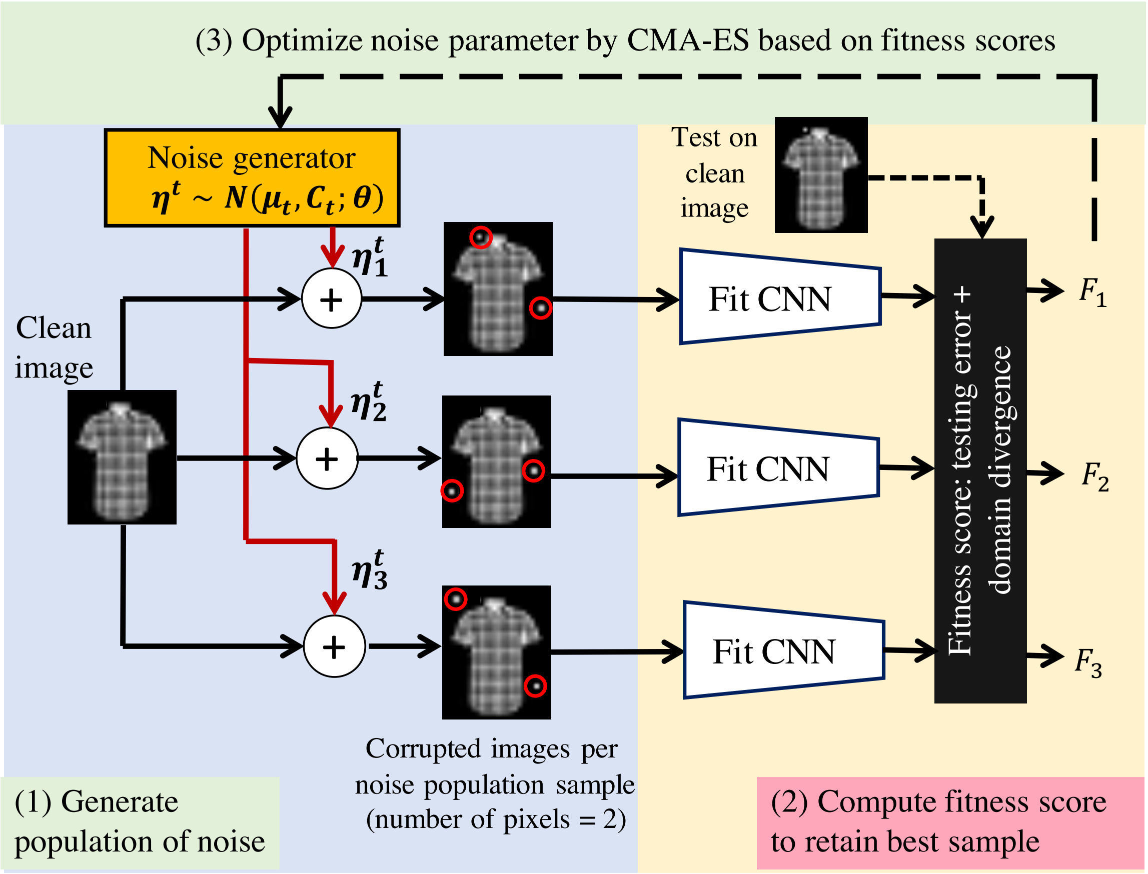
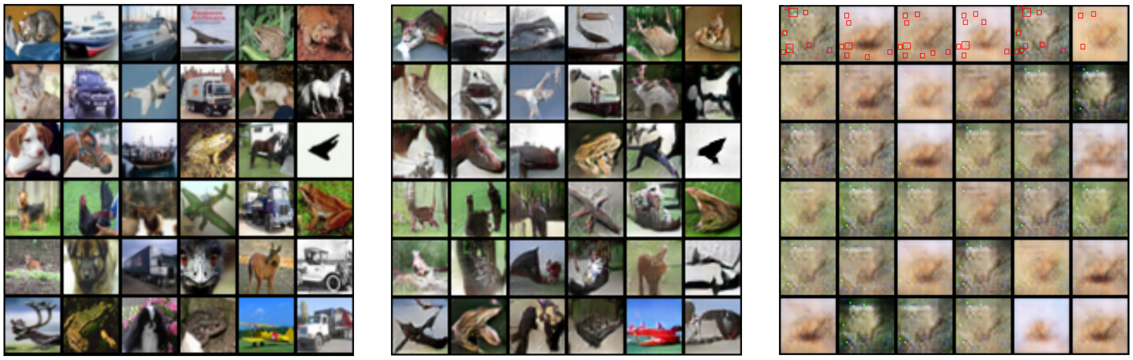

(3) Optimize noise parameter by CMA-ES based on fitness scores
Noise generator
$\eta^t \sim N(\mu_t, C_t; \theta)$
Test on clean image
Clean image
η_1^t
η_2^t
η_3^t
+
Fit CNN
Fitness score: testing error + domain divergence
F_1
F_2
F_3
Corrupted images per noise population sample (number of pixels = 2)
(1) Generate population of noise
(2) Compute fitness score to retain best sample

Figure 2-1: Overview of our proposed evolutionary algorithm for training time attack optimization. We sample from a noise generator ($N_p = 2$ case shown) to perturb a few pixels in training images and fit CNN models on noisy data. Evaluation is performed on clean data from the training set to find fitness scores (high generalization loss). Noise generator parameters are updated by evolutionary strategy [40].

perturbation based methods fall under the category of evasion attacks that fools the model during inference.

This work introduces a novel concept of adversarial training time attack, which we call as *EvoShift*, defined as a malicious change in training images only limited to a few pixel changes. We propose a novel loss function to design such pixel changes. CNNs trained on such images will cause low test accuracy on clean images from the test set leading to a drastic drop in neural network generalization. Figure 2-1 shows an overview of our method. Our method has some similarity to data poisoning attacks [12,91,96], where the adversary injects a few malicious samples in the training data to cause incorrect classification (typically targeted) during inference. However,



(a) Samples from true image distribution.
(b) GAN reconstruction on true images.
(c) GAN reconstruction on our attacked data.

Figure 2-2: Illustrating our proposed attack method in GANs: (a) true images on CIFAR10 dataset, (b) images generated by GAN [57] trained on the true data distribution, (c) images generated by training on our proposed attacked data by just changing only 10-pixel values. Generated images faithfully recreate the pixel nuisance features (highlighted in the first image row with red markers) while ignoring the semantic features like object shape and color.

our method is technically different from poisoning attacks. Poisoning methods only target a few images in the test set to attack by changing the decision boundary in a limited local region. In contrast, our method's main objective is to induce overfitting in neural networks. The proposed attack tries to change training images so that the induced decision function shows a significant departure from the true decision boundary. Our attack finds the best location of pixel disturbance for each class in a multi-label dataset that maximally increases overfitting. Using our method, we expose serious vulnerabilities in neural networks that can overfit to even single-pixel disturbance which is an undesirable feature for robust machine learning.

We propose a novel evolutionary strategy based algorithm, called *EvoShift*, for optimizing pixel attacks that are added to the training images. Our contributions in this work can be summarized as

- To obtain our training time attack, we solve a joint $\min\max$ optimization with the outer maximization designed to find the pixel noise and inner minimization designed to train the neural network on the noisy images. We impose a constraint on the optimization such that the cross-entropy (CE) loss on the noisy images is low, and the loss on the clean images is high. Figure 2-1 shows the various components of our proposed algorithm for finding optimal training pixel attack. Such a formulation results in CNN trained on the noisy images to have a very high error on clean test images, thus exposing serious vulnerabilities in CNNs that are detrimental to robust learning. Figure 2-1 shows an overview of our proposed training time attack.

- Interestingly, we find that optimization choice plays a vital role in generalization robustness. We show empirical evidence that SGD is resilient to our training time attacks, unlike adaptive optimization techniques (Adam). Although adaptive optimization methods are a popular choice for practitioners and researchers, we show that they can easily overfit in the presence of our training time attacks. We believe that this is an important finding for the machine learning research community.

- We also apply regularization methods to counteract our proposed adversarial training time attack and find that most well-known regularization methods are ineffective against our attack. We find that random-crop data augmentation is moderately effective for a few pixel attacks.

- As a defense against our attack, we propose a novel robust loss function for CNN classification training that is resilient against our training attacks using an information maximization framework. This result suggests that the traditional cross-entropy minimization framework for CNN training might cause non-robust feature learning, which might be mitigated by our proposed information-theoretic loss function.

- We introduce the concept of vulnerability in GANs under proposed *EvoShift* attacks, causing poor image generation quality due to overfitting in the GAN discriminator. Figure 2-2 shows that GANs trained under our proposed attack fails to obtain a detailed reconstruction of the object in the image, thus exposing weakness in image generation using GANs. To the best of our knowledge, this is the first work showing vulnerabilities in GANs under training time attacks.

Additionally, we proposed the concept of adversarial training time attack for GANs (Section 2.5.4) and introduced robust loss function for CNN learning (Section 2.7). We also performed additional experiments for partial dataset attack (Section 2.8.3), training accuracy in the presence of regularization (Section 2.8.3), and transfer of our attack to ImageNet dataset using Spatial Value Function (SVF) method (Section 2.5.5).

2.2 Related works

Adversarial attacks: This recent line of work [6, 16, 63, 98] demonstrated that it is possible to fool trained neural networks using *adversarial* query images that are imperceptible from normal unperturbed images. Su et al. [97] showed that it is possible to craft adversarial test images by single-pixel perturbations in training images. These attacks fall under the category of *evasive* attacks that exploit the weakness in trained models by attacking query images. Instead of attacking query data during inference, our method corrupts the training data to maximize the generalization error.

Data poisoning: In data poisoning, the attacker injects malicious samples in the training data to control the model behavior during test time. Such an attack was first introduced in Support Vector Machines (SVM) for binary classification problems in [12]. Recently, there have been some works in the field of neural networks [96] as well. Koh et al. [55] used influence functions to synthesize adversarial training examples that can flip the predicted labels of a set of testing images. Shafahi et al. [91] used a forward-backward-splitting iterative procedure [32] to create targeted data poisoning attacks that performed better than previous methods. As we mentioned earlier, different from previous works, our method presents a general gradient-free strategy for crafting adversarial training perturbations, which is agnostic to the underlying learning algorithm, with precise control on noise parameters. Jacobsen et al. [48] studied the effect of single-pixel perturbations on MNIST training images on test performance. They showed that adding one pixel to training images that encodes the class label, and then testing on the clean test set, can yield a high generalization gap. Tanay et al. [99] showed that neural network models could be made almost arbitrarily sensitive to a single-pixel while maintaining identical test performance between models. Unlike poisoning methods, our method's main objective is to induce overfitting in neural networks using the proposed gradient-free optimization.

Neural network generalization: Numerous previous works [41, 70, 78, 106] studied the generalization properties of neural networks under such high complexity parameter space. Zhang et al. [113] showed that neural networks could fit random noise. The idea of pixel perturbation has also been explored in [114] to measure the testing accuracy of images. Unlike previous works, our method analyzes the robustness of neural networks under optimally crafted perturbations in training images, similar to Wilson et al. [105], which presented a manually crafted artificial example. However, our method is a generalization to such problems that can generate optimal training perturbations for an arbitrarily sized dataset using evolutionary algorithms.

Information theory-based methods: We also review some generative learning methods for adversarial defense. Alemi et al. [4] showed that learning with Variational Information Bottleneck (VIB) is robust to standard perturbation based adversarial example. Song et al. [94] proposed a generative model called PixelDefend to detect adversarial samples and moving them back to the training data distribution. Meng et al. [64] used autoencoders to detect adversarial inputs by using the reconstruction threshold and proposed a mechanism to defend against a gray box attack. With the recent interest in the information-theoretic view because of the information bottleneck [92, 102], the estimation of mutual information [7, 9] has attracted a lot of attention.

2.3 Problem Formulation

We consider a multi-class classification task with input space $X \in \mathbb{R}^N$ and label space $Y = \{1, ..., N_c\}$. The true data distribution is given as, $S = \{\boldsymbol{x}_i, y_i\}_{i=1}^n \sim \mathcal{D}_S$, where n is the total number of images, $\boldsymbol{x}_i$ is an instance of the image from the dataset and y_i is the corresponding image label. Our goal is to design an pixel-perturbation attack such that the classifier trained on the perturbed training data yields high empirical risk (or test error) on the true uncorrupted samples. Only the training set is used for obtaining such pixelwise attack. Considering that for each sample in S, we can draw class-wise input perturbations, $\boldsymbol{\Delta} = \{\boldsymbol{\delta}_i\}_{i=1}^{N_c} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, parameterized by the mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. N_c is the total number of classes. The class-wise noise is added to training image as $\boldsymbol{x}_i^p = \boldsymbol{x}_i + \boldsymbol{\delta}_{y_i}$. The joint distribution of the perturbed data, which is constructed by assigning labels of the true samples to the corresponding perturbed samples, is given as $P = \{\boldsymbol{x}_i^p, y_i^p\}_{i=1}^n \sim \mathcal{D}_{adv}$.

Let us define a classifier function $h : X \to Y$ from a hypothesis space $\mathcal{H}$. The corresponding empirical risk on samples drawn from a distribution $\mathcal{D}$ is defined as, $R_{\mathcal{D}}(h) \overset{\text{def}}{=} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\big(I[h(\boldsymbol{x}) \neq y]\big)$, which signifies the error on the samples drawn from $\mathcal{D}$. Our objective is to find optimal perturbation parameter that increases the empirical risk on the clean samples while minimizing it on the corrupted samples, thus compromising generalization in neural networks. This is given as

$$\begin{gathered}\boldsymbol{\theta}^* = \max_{\boldsymbol{\theta}}\Bigg(R_{\mathcal{D}_S}(h^*) - R_{\mathcal{D}_{adv}}(h^*)\Bigg) \\ s.t.\ h^* = \arg\min_{h\in\mathcal{H}} R_{\mathcal{D}_{adv}}(h).\end{gathered} \tag{2.1}$$

The above objective finds an optimal perturbation parameter that increases the empirical risk on the clean samples while minimizing it on the corrupted samples, thus compromising generalization in neural networks.

2.4 Theory on Maximum Domain Divergence based Perturbation Optimization

This section outlines the theory behind domain divergence-based perturbation optimization, which lays the foundation for our evolutionary strategy-based perturbation optimization.

2.4.1 Domain divergence

Given a source domain ($\mathcal{D}_S$) and target domain ($\mathcal{D}_T$), the notion of domain divergence refers to how samples in each of the domains differ from the other. For the conventional risk minimization regime, "domain gap" can be measured by the difference between empirical risk in the source and target domains. Shai et al. [10, 11] formally defined this notion as the proxy $\mathcal{A}$-distance, which was used by domain adaptation methods [3, 30] for reducing the source and target domain errors in an adversarial setting.

Let us consider a domain $\mathcal{X}$ and a collection of subsets of $\mathcal{X}$, given as $\mathcal{A}$. Given two domain distributions $\mathcal{D}_S$ and $\mathcal{D}_T$ over $\mathcal{X}$, and a hypothesis class $\mathcal{H}$, the $\mathcal{A}$-divergence between the domains is given as

$$d_{\mathcal{A}}(S,T) \overset{\text{def}}{=} 2 \sup_{A \in \mathcal{A}} \big| \Pr_{\mathcal{D}_S}[A] - \Pr_{\mathcal{D}_T}[A] \big|, \tag{2.2}$$

where the hypothesis class $\mathcal{H}$ is a class of functions representing binary classifiers and is symmetric as defined in [11]. The above distance is the $\mathcal{H}$-divergence $d_{\mathcal{H}}(.,.)$ when we compute the distance of the class of subsets with characteristics functions in the hypothesis space $\mathcal{H}$.

Shai et al. [11, 30] showed that although it is generally difficult to compute the $\mathcal{H}$-divergence for the hypothesis space of linear classifiers, it can be approximately computed using the empirical $\mathcal{H}$-divergence from samples $\boldsymbol{x}_i^s \sim \tilde{D}_S$ and $\boldsymbol{x}_i^t \sim \tilde{D}_T$, and is defined as

$$\hat{d}_{\mathcal{H}}(S,T) \overset{\text{def}}{=} 2\left(1 - \min_{h \in \mathcal{H}} \left[\frac{1}{n}\sum_{i=1}^{n} I[h(\boldsymbol{x}_i^s)=0] + \frac{1}{n'}\sum_{i=n+1}^{N} I[h(\boldsymbol{x}_i^t)=1]\right]\right), \tag{2.3}$$

where n samples from the source domain and n' samples from the target domain are drawn. Given samples from the two domains, the above empirical distance can be computed as the proxy $\mathcal{A}$-distance by learning a classifier $h \in \mathcal{H}$, which optimally learns to discriminate between the source and target samples. The proxy $\mathcal{A}$-distance is defined as, $\hat{d}_{\mathcal{A}} = 2(1-2\epsilon)$ according to [11], where ϵ is the discriminator error.

2.4.2 Bound on target risk

We are interested in finding a bound of the empirical target risk obtained by learning a source samples classifier. Shai et al. (and later used by Ganin et al. [10, 11, 30]) showed that the bound on the target risk could be computed in terms of the proxy $\mathcal{A}$-distance defined above, as follows

Theorem 2.4.1. *Considering $\mathcal{H}$ be a hypothesis class of Vapnik–Chervonenkis (VC) dimension d, for n samples $S \sim (\tilde{D}_S)^n$ and $T\tilde{(}\hat{D}_T)^n$, then with probability $1-\delta$ over the choice of samples, for every $h \in \mathcal{H}$:*

$$\begin{aligned}\hat{R}_T(h) \leq \hat{R}_S(h) + \sqrt{\frac{4}{n}\left(d\log\tfrac{2e\,n}{d} + \log\tfrac{4}{\delta}\right)} \\ +\hat{d}_{\mathcal{H}}(S,T) + 4\sqrt{\frac{1}{n}\left(d\log\tfrac{2n}{d} + \log\tfrac{4}{\delta}\right)} + \beta,\end{aligned} \tag{2.4}$$

with $\beta \geq \inf\limits_{h^* \in \mathcal{H}} \left[R_S(h^*) + R_T(h^*)\right]$ and $\hat{R}_S(h) \;=\; \frac{1}{n}\sum_{i=1}^{m} I\left[h(\boldsymbol{x}_i^s) \neq y_i^s\right]$.

Given a fixed hypothesis space, we observe that increasing the $\mathcal{H}$-divergence between the two domains would make the above bound loose. It is to be noted that high domain divergence increases the range of values for target risk, increasing the likelihood of overfitting.

The above analysis is relevant in our setup since we are interested in finding perturbations that, although constrained to a few pixel changes, increase the generalization gap (analogous to $\mathcal{H}$-divergence) between clean and perturbed training images. Although the above analysis is shown for a binary classification system, it is relevant to multi-class classification systems. We use this insight in our formulation to craft a fitness score to increase the domain divergence between the true and perturbed distributions.

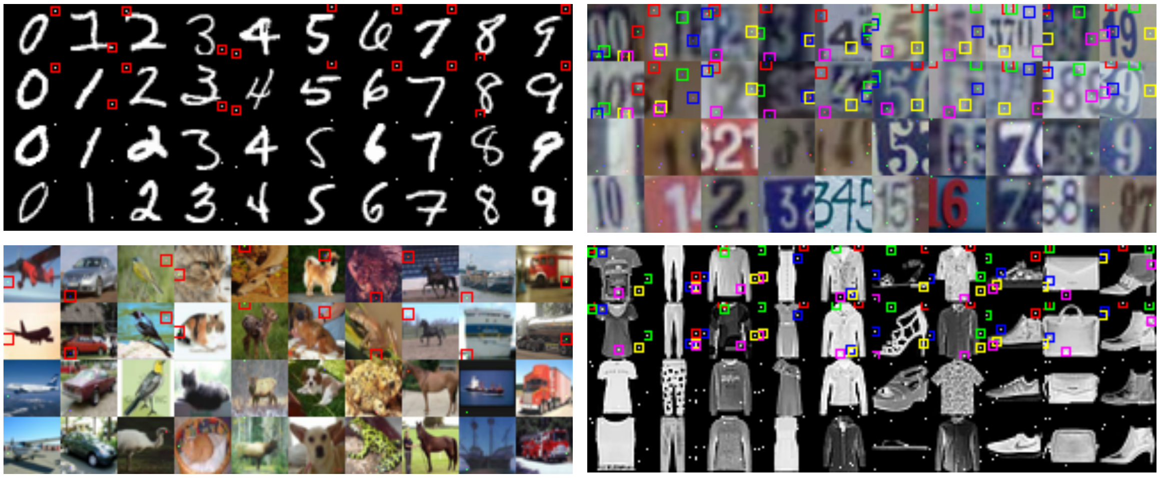


Figure 2-3: Perturbed image sample for single-pixel attack by our proposed EvoShift algorithm, showing class-wise pixel perturbations for (a) MNIST (top-left), (b) SVHN (Top-right), (c) CIFAR10 (Bottom-left), and (d) Fashion-MNIST (Bottom right). The top two image rows in each dataset samples are highlighted to help the reader spot the classwise pixel perturbations.

2.5 Our Proposed Pixel-based Perturbation

Based on the domain divergence theory, we outline our proposed noise optimization strategy in this section. First, we explain how we parametrize the noise, then we describe our cost function, and finally, we present our proposed algorithm for optimal perturbation generation.

2.5.1 Parameterizing pixel attack

We design our adversarial training time attack in the form of few-pixel perturbations for each class. Let us assume there is a total of N_c classes in the dataset. We perturb images of the same class label, with N_p pixel perturbations, which is represented as $\mathbf{\Delta} = \{(x_0^0, y_0^0, v_0^0),\ \ldots\ (x_{N_p}^0, y_{N_p}^0, v_{N_p}^0),\ \ldots(x_0^{N_c}, y_0^{N_c}, v_0^{N_c}),\ldots(x_{N_p}^{N_c}, y_{N_p}^{N_c}, v_{N_p}^{N_c})\}$, consisting of $(N_p \times N_c)$ pairs of pixel noise (x, y, v) where (x, y) represents the spatial locations of pixel noise and v represents the intensity of pixel disturbance.

Given a noise sample $\mathbf{\Delta}$, all images I_j in class k will have their pixel value represented by (x_i^k, y_i^k) assigned the intensity value v_i^k, such that, $I_j[x_i^k, y_i^k] = v_i^k$, for i=1 to N_p. These pixel perturbations act as the distractor features in our training images.

The motivation behind class-wise pixel attack encoding is to force the neural network to use the noisy pixel as the discriminative feature for that class while ignoring the semantic features like image appearance and color information. Our goal is to find the distribution of spatial locations where such pixel distractions are most effective for overfitting the cross-entropy (CE) loss.

During optimization, we draw N_p pixel perturbations for each class, from a multivariate normal distribution, $\boldsymbol{\Delta} = \{\boldsymbol{\delta}_i\}_{i=1}^{N_C} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, parameterized by $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ which are the mean vector and the covariance matrix respectively. $\boldsymbol{\Delta}$ represents the $3 \times N_C \times N_p$ dimensional parameterization vector comprising of all class-wise perturbations. For color images, $\boldsymbol{\Delta}$ has $5 \times N_C \times N_p$ dimensions because the pixel value $v = (v_R, v_G, v_B)$ consists of three values for each channel. The optimization of these pixel noise is explained in the following section.

2.5.2 Cost Function

Our neural network model F_θ is trained on the adversarially attacked training data by optimizing the CE loss, using the traditional training objective, $\mathbb{E}_{(x,y)\sim\mathcal{D}_{adv}}\big[\mathcal{L}_{CE}(F_\theta(x), y)\big]$. We propose a perturbation objective that will be successful if the model F_θ has low CE loss on the perturbed training images and high loss on the clean images. We combine the above condition in the form of a single equation, given as

$$\max_{\boldsymbol{\Delta}, s=0} \min_{\theta, s=1} \mathbb{E}_{(x,y)\ \mathcal{D}}\big[\mathcal{L}_{CE}(x + s\boldsymbol{\Delta}, y; \theta)\big]. \tag{2.5}$$

In the above equation, s acts as a switch to turn on/off the training data's perturbation. The minimization concerning the neural network parameters θ is performed in the presence of the perturbation, such that it overfits the noise. The maximization is performed with $s = 0$ such that the CE loss on clean samples should be high, thus creating an adversarial attack that maximizes the generalization gap encouraging overfitting.

According to Equation 3.12, it is difficult to optimize the noise parameter $\boldsymbol{\Delta}$ using standard gradient-based methods, because the gradient concerning $\boldsymbol{\Delta}$ is 0, due to multiplication with s. Therefore, we resort to gradient-free Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES) based optimization for finding the optimal perturbations. We describe the cost function for our CMA-ES optimization consisting of the following components as follows.

Semantic mismatch cost: Using the above argument, our fitness score should encourage high cross-entropy loss on the images from true data distribution while showing a small loss on the adversarially attacked training samples. The above condition emulates overfitting in the model. This is formulated as the difference of loss terms between these scenarios which we designate as the semantic mismatch cost (S_m) as follows

$$S_m = \frac{1}{N} \sum_{(x,y)\sim\mathcal{D}} \Big[\mathcal{L}_{CE}(x, y; \theta) - \mathcal{L}_{CE}(x + \mathbf{\Delta}, y; \theta) \Big], \tag{2.6}$$

where the above score is maximized by the CMA-ES for a fixed θ trained on $(x+\mathbf{\Delta}, y)$ samples. The first term encourages a high loss on samples drawn from the true distribution, while the second term promotes a low loss on the perturbed image. This score measures the generalization gap between the samples drawn from true distribution and perturbed distribution which differ by only a few pixels.

Domain divergence: Given a source domain $\mathcal{D}_S$ and target domain $\mathcal{D}_T$, the notion of domain divergence refers to how samples in each of the domains differ from the other, as explained in Section 2.4. In our settings, we want to train the model to have low empirical risk on samples from $\mathcal{D}_{adv}$ and high risk on the true distribution $\mathcal{D}$. This can be viewed as an increasing domain divergence between these distributions. Shai et al. [11] and Ganin et al. [30] showed that approximate domain divergence could be computed by learning a binary classifier $h \in \mathcal{H}$, which optimally learns to discriminate between the source and target samples. In our case, we train a discriminator between uniformly sampled images from the true (label 1) and attacked (label 0) distributions. The domain divergence score S_d is computed as

$$S_d = 2\Bigg(1 - \big[\mathbb{E}_{x\in\mathcal{D}}\mathbb{I}[h(\boldsymbol{x})=0] + \mathbb{E}_{x\in\mathcal{D}_{adv}}\mathbb{I}[h(\boldsymbol{x})=1]\big]\Bigg), \tag{2.7}$$

where $\mathbb{I}(.)$ is the indicator function used for computing the error in discriminator. Intuitively, for attacked training samples in a population (for CMA-ES) that are dissimilar from the true samples, the discriminator can learn good separable features, thus having high domain divergence. We empirically found that by adding the domain divergence score, the stability of convergence for the CMA-ES algorithm can be improved; however, the final fitness score achieved is comparable to when it is not used.

2.5.3 Our Proposed EvoShift algorithm

Based on the above-discussed theory and cost function analysis, we present the details of the evolutionary strategy based adversarial attack algorithm (EvoShift) as explained in Algorithm 3. We start the first generation of CMA-ES from initial perturbation parameter $\boldsymbol{\theta}_0 = (\boldsymbol{\mu}_0, \boldsymbol{C}_0)$. For each generation t, we sample multiple pixel perturbation parameters $\{\Delta_j\}_j$ and obtain the optimal neural network weights $\boldsymbol{\theta}^*$, by training a CNN from scratch on each such perturbation samples by minimizing the cross-entropy loss. After each generation, the sampling parameters are updated by the CMA-ES algorithm to retain the attacks corresponding to the top-performing costs [1]. The attack corresponding to the best performing cost across all generations is returned. We find that models trained on such samples show poor generalization, thus uncovering significant vulnerabilities in CNNs. It is to be noted that only samples from the training set were used to optimize the attack in the above algorithm. No test set samples were seen during attack optimization or model training. Figure 3-1 shows our proposed attacked training data on various datasets.

2.5.4 Extension to Attacks on Generative Adversarial Networks

The above attack analysis is targeted toward multi-class classification systems. However, this can also be used for attacking generative models that consist of a discrim-

[1] More details on the CMA-ES algorithm can be found in the original paper [40].

Algorithm 1 EvoShift

```
Require: Training data $(x, y) \sim \mathcal{D}$, ES params $\boldsymbol{m}_0, \boldsymbol{\Sigma}_0, \sigma_0$
for $t$ from 0 to $N_{gen}$ do
    Sample a population of noise: $\{\boldsymbol{\Delta_j}\}_{j=1}^{\lambda} \sim N(\boldsymbol{\mu_t}, \boldsymbol{\Sigma}_t)$, where $\lambda$ is population size in a generation.
    Fit $j^{th}$ models: $\min_\theta \mathbb{E}_{(x,y)\sim\mathcal{D}}[\mathcal{L}_{CE}(x + \boldsymbol{\Delta_j}, y; \theta)]$
    Get the $j^{th}$ semantic score as $\mathcal{S}_m^j = \frac{1}{N}\sum_{(x,y)\sim\mathcal{D}}\left[F_\theta(x)_y + \sum_{j\neq y}(1 - F_\theta(x)_j)\right]$
    Train discriminator to classify between the true samples $x$ and attacked samples $x + \delta_j$, and assign domain divergence cost as $\mathcal{S}_d^j = -2(1 - 2\epsilon)$
    Compute total cost for the $j^{th}$ sample $\mathcal{C}_j = \mathcal{C}_m^j + \mathcal{C}_d^j$.
    Update ES parameters based on the fitness score $m_{t+1}, \boldsymbol{\Sigma}_{t+1}, \sigma_{t+1} =$ CMA-ES$(m_t, \boldsymbol{\Sigma}_t, \sigma, \{\mathcal{C}_j\}_j)$.
    Store the solution with best fitness score in $\boldsymbol{\Delta}^*$
end for
return best solution: $\delta^*$ as output
```

inative subsystem, namely, Generative Adversarial Network (GAN) [34]. GANs are generative models that learn the data distribution of the training data, which can then be used for creating novel samples from it. It consists of a generator network (G) and a discriminator network (D) and a learning objective function consisting of a min-max optimization as shown below

$$\min_{G} \max_{D\in(0,1)} \mathbb{E}_{x\sim p_{data}} \log(D(x)) + \mathbb{E}_{z\sim p_z} \log(1 - D(G(z))), \tag{2.8}$$

where the discriminator classifies between samples from the generator and true data distribution, providing a gradient signal for the generator to produce samples similar to the data distribution. Typically, the discriminator is trained using CE loss to classify between the true image samples and generated samples, although some recent works use a different loss for discriminator learning [5,37]. In this work, we specifically use the implementation of [57], which uses CE loss for discriminator training, which is a binary classifier in an encoder-decoder setting. We show that our proposed training time attack can also cause overfitting on the GAN's discriminator resulting in poor reconstruction of images, as shown in Figure 2-2 due to suppression of the semantic features.

2.5.5 Extension to Attacks on Larger Datasets

In the above sections, we discussed the proposed attack being optimized for a particular dataset. Computing the optimal pixel attack requires training multiple CNNs on a small subset of the dataset for each generation, which can be computationally expensive for limited resources. Therefore we propose a method to transfer the learned pixel location from a smaller dataset to a larger dataset without using the expensive gradient-free attack optimization. We call this method Spatial Value Function (SVF) sampling. The idea is to sample pixels close to the attack locations of the source dataset and apply that to the target dataset. We compute the SVF by convolution with a Gaussian Point Spread Function (PSF) at the pixel perturbation location(s) for the source dataset as follows

$$\mathrm{SVF}_{t+1}(x,y) = \sum_{j} \sum_{(x_k,y_k)\in\delta_j} \mathrm{PSF}(x-x_k, y-y_k), \tag{2.9}$$

where the index j iterates over all the classes and k iterates over the pixel corruptions in each class (depends on N_p). We extend the SVF to match the target image dimensions by performing bilinear interpolation. Finally, the pixel corruption location on the target dataset is obtained by importance sampling based on the reshaped SVF on the target dataset. We assume that the source dataset and the target dataset have a spatially similar primary object location. We consider CIFAR10 as the source dataset. For the target dataset, we choose a subset of 64×64 ImageNet dataset with ten classes. For these two datasets, it is reasonable to assume that the object would be primarily located at the image center. We show that such transferred attacks to the ImageNet data successfully reduces testing accuracy in Section 2.8.11.

In case the number of classes in the target dataset is more than that of the source dataset, we can sub-sample within the spatial value function to sample the required number of pixel attacks. For example, if the target dataset has C_t number of classes whereas the source has C_s classes, where $C_t > C_s$, we can create C_t small density distribution on the SVF using techniques like Expectation Maximization [**?**]. Following that, sampling from each such small distribution would give us the pixel locations for the attack pixels. However, in order to ensure non-overlapping distributions, we also have to maximize the entropy of the overall system. This method would give well-separated distributions if C_t and C_s are not separated by large orders of magnitude, which otherwise would lead to overlapping pixel attacks for multiple classes. In such cases, intermediate fine-tuning using the proposed evolutionary strategy might be required to ensure good separability of attack pixels.

2.6 Explaining Poor Generalization to Proposed Pixel Attack

In this section, we identify that the drastic drop of generalization performance in CNNs for classification models [42, 46] and GANs [34] under our proposed training time attack, which is typically due to the over-reliance of cross-entropy loss on the added noise in the image. Traditional training with cross-entropy loss results in unconstrained mutual information maximization between the learned features and the target labels, resulting in over-dependence on attack pixel features, even if they are not semantically meaningful. We propose a robust feature learning scheme that preserves the semantic information by maximizing the mutual information between the latent features and input images to mitigate this problem.

Considering x as the input image and z as the latent features (logits), which is computed from the deep model as $z = F_\theta(x)$, the goal of CE loss is to maximize the mutual information between the features and the target labels y. Such formulations are typical in image classification networks and GANs for discriminator learning.

Given a data distribution $(x, y) \sim \mathcal{D}$, the goal of the classification network is to maximize the mutual information $I(y; z)$ as stated in [48]. However, in the absence of priors, such methods can learn highly predictive features that do not align with the human perceptual system. Specifically, in our EvoShift algorithm, our noisy perturbations in the training images act as highly discriminative features, which leads to suppression of semantic features. From an information-theory point of view, the cross-entropy loss does not preserve the source distribution information. It only focuses on the high predictive features that lead to vulnerabilities in the presence of our proposed training time attack.

2.7 Robust Feature Learning

In this section, we propose a novel loss function for training a CNN that is robust against overfitting to spurious pixel perturbations (similar to our proposed attack). We also specify metrics to measure the model's affinity to learn features based on noisy pixel artifacts ("nuisance") or object properties like shape and color ("semantics").

2.7.1 Robust Training with Mutual Information Constraints

In the presence of our proposed perturbed training samples, neural networks overfit to the spurious features leading to over-reliance on such spurious features. This results in an invariant response in the presence of such artifacts, which is termed as semantic invariance by [48]. The high semantic invariances induced in the neural network models can be attributed to the standard cross-entropy loss function's insufficiency, which favors choosing simple predictive features for the label rather than complicated features that require multi-layered reasoning. Therefore, under our proposed EvoShift, CE loss learns a decision function based on the spurious input perturbations.

To alleviate this issue, Jacobsen et al. [48] designed a bijective neural network model that preserves the input information, thus capturing all variations in the input. However, standard neural network classifiers are not bijective by design. Therefore there might be a loss of useful semantic information in such networks. Inspired by the concept of information preservation, we present a robust feature learning schematic that introduces an additional constraint in the objective function and the standard cross-entropy loss. In addition to maximizing mutual information between the feature and the labels, learned features should maximize the mutual information $I(z;x)$ between the feature and image. This forces the latent representations to preserve class attributes like shape and appearance, which is used to reconstruct the image features. Thus, the modified objective function can be formulated as

$$\max_{\theta}[-\mathcal{L}_{CE}(y, z; \theta) + I(z, x; \theta)] \quad s.t. \;\; I(z, x; \theta) < I_c, \tag{2.10}$$

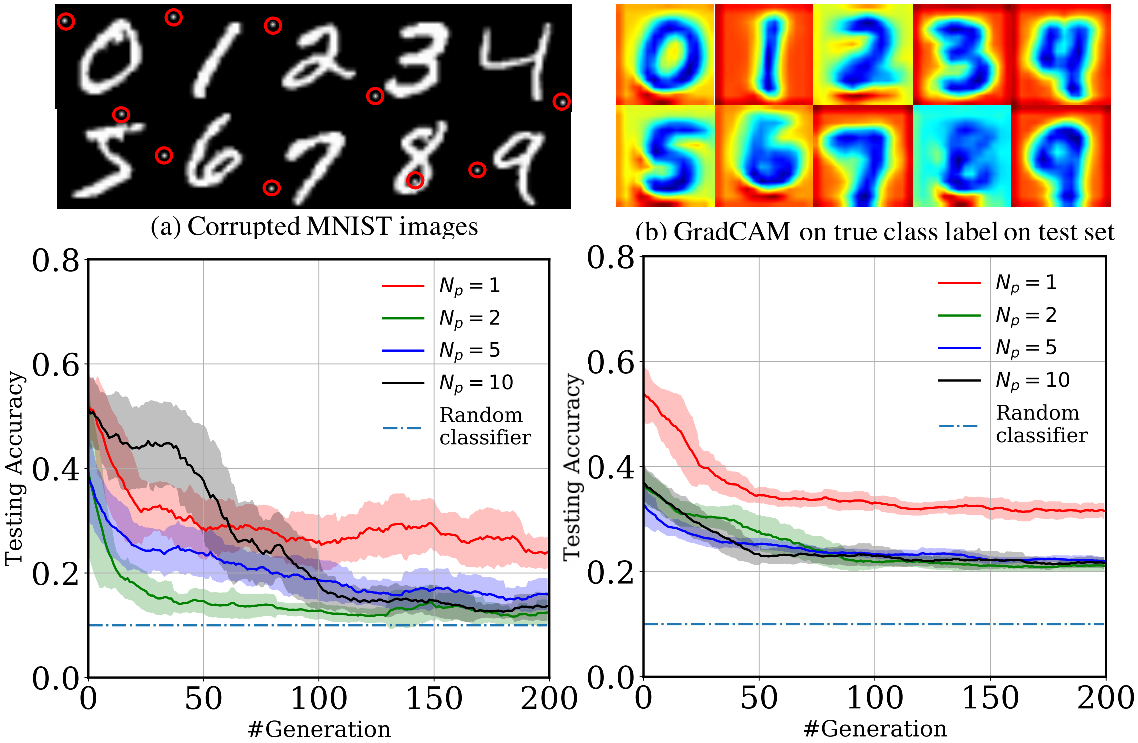
where I_c is a bound on the mutual information without which we obtain the trivial solution $z = x$. The objective can be solved using the Lagrangian multiplier. Since it is intractable to find the marginal distribution $p(z)$ for the above objective, we minimize an upper bound of the regularized objective using an approximation of the marginal $r(z)$ following [4]. The robust objective can be written as

$$\begin{aligned} J(\boldsymbol{\theta}_E, \boldsymbol{\theta}_G, \boldsymbol{\theta}_C) = \mathbb{E}_{\boldsymbol{z} \sim E_{\theta_E}(\boldsymbol{z}|\boldsymbol{x})} \Bigg[\overbrace{\sum_{i=1}^{C} -y_i D_{\theta_C}(\boldsymbol{z})_i}^{\text{Cross-Entropy Loss}} - \overbrace{\log q_{\theta_G}(\boldsymbol{x}|\boldsymbol{z})}^{\text{Decoder reconstruction loss}} \Bigg] + \\ \beta \mathbb{E}_{\boldsymbol{x} \sim p(\boldsymbol{x})} \underbrace{[\mathrm{KL}[E_{\theta_E}(\boldsymbol{z}|\boldsymbol{x}) || r(z)]]}_{\substack{\text{KL divergence between} \\ \text{encoder and marginal}}}. \end{aligned} \tag{2.11}$$

In the above loss function, we simultaneously optimize an encoder E_{θ_E}, a generator q_{θ_G}, and a classifier D_{θ_C}. This is implemented in practice by a Convolutional Variational Autoencoder [54] with the encoder and decoder networks. The classifier network D_{θ_C} is trained on the latent code from the bottleneck layer. During inference, the mean latent code is used as the feature for the classifier. We refer to this proposed loss function as "vibCE", due to its analytical similarity to the Variational Information Bottleneck (VIB) [4] based methods.

2.7.2 Robustness Evaluation Metrics

As explained in the previous section, there is a need to disentangle such factors of variations for systematic quantification of semantic features suppression by our proposed adversarial training time attack. Standard accuracy metrics cannot separate the effect of semantic and spurious features on classification performance. Therefore, we introduce two disentangled metrics, addressing the influence of semantic features



(a) Corrupted MNIST images

(b) GradCAM on true class label on test set

(c) Learning curve: MNIST

(d) Learning curve: Fashion-MNIST

Figure 2-4: (a) Highlighting learned single pixel perturbations on MNIST images, (b) GradCAM visualization of the last Conv layer for $N_p = 1$. Dominant gradient distribution is in the background. Learning curve with increasing generations of CMA-ES is shown for (c) MNIST and (d) Fashion-MNIST.

and spurious task-irrelevant feature on classification performance, which are described below.

Semantic Sensitivity (α_S): In the presence of predictive features due to our proposed training time pixel attack, the model learns spurious highly predictive features for encoding class information. However, a robust learning objective should learn to ignore such spurious features while focusing on semantic features like image shape and appearance. This metric measures the contribution of task-relevant features toward classification performance by computing the test accuracy on clean data distribution in the absence of predictive nuisance features. A robust classifier would produce test accuracy close to 1.0, even in the presence of our proposed pixel-wise attack. In contrast, compromised models will have test accuracy $\approx \frac{1}{N_c}$ (which is same as a random classifier). We define semantic sensitivity as

$$\alpha_S = \frac{\overbrace{\mathbb{E}_{(x,y)\in\mathcal{D}}\mathbb{I}[F_\theta^\delta(\boldsymbol{x})=y]}^{\text{test accuracy on clean data}} - \frac{1}{N_C}}{1.0 - \frac{1}{N_C}}, \tag{2.12}$$

where F_θ^δ refers to the classifier trained on attacked data. We normalize α_S to the range of $[0, 1]$. A robust classifier should have α_S close to 1.0.

Nuisance Sensitivity (α_N): This metric measures how easily a model can overfit in the presence of spurious features in the attacked training data. A robust model should produce the same network response irrespective of whether the true images are attacked or not. For example, if the pixel perturbation for class 9 is overlayed on images from an arbitrary class and the model predicts the label 9 for most images, it has high nuisance sensitivity. To measure this, we overlay each class-specific noise on all test images and measure the accuracy for that class over all classes. Thus, we define nuisance sensitivity (normalized between $[0, 1]$) as

$$\alpha_N = \frac{\overbrace{\mathbb{E}_{k\sim\text{Unif}(1,\ldots,N_C)}\mathbb{E}_{(x,y)\in\mathcal{D}}\mathbb{I}[F_\theta^\delta(\boldsymbol{x}+\boldsymbol{\delta_k})=k]}^{\text{test accuracy on noise overlayed images}} - \frac{1}{N_C}}{1.0 - \frac{1}{N_C}}, \tag{2.13}$$

where $\text{Unif}(1, ..., N_c)$ is the uniform sampling of class labels and $\mathbb{I}$ is the indicator function which counts the number of images that are classified as the attack class k. Robust classifiers should have α_N close to 0.0 because it should not respond to nuisance features that is added by our training time attacks.

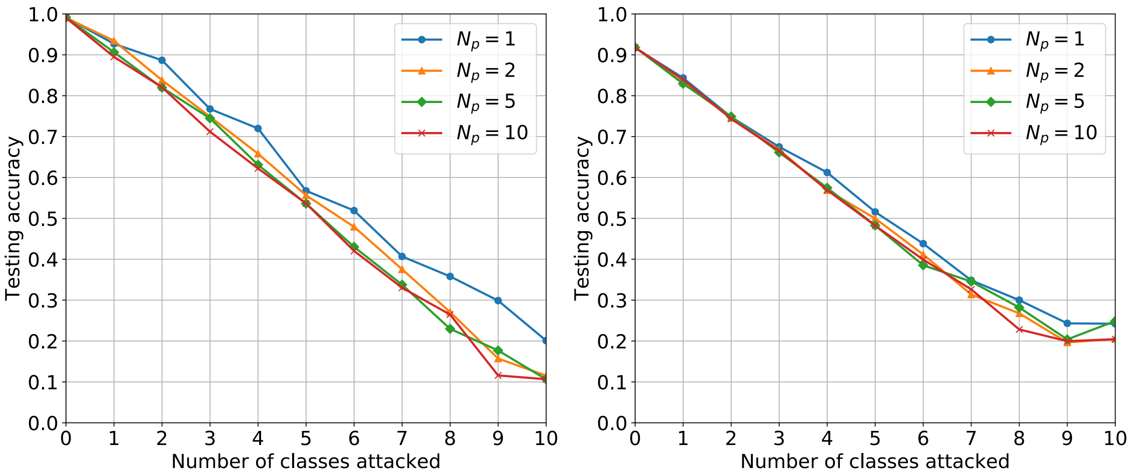
2.8 Experimental Results

In the experimental section, we perform extensive empirical analysis to address the following questions: (a) Can our proposed algorithm learn optimal noise configuration, which is better than randomly perturbed noise?, (b) Can our method outperform previous training time attack methods?, (c) How does our method affect partial attack on training data?, (d) Can popular regularization techniques defend against such attacks?, (e) Is our proposed attack effective in zero-shot transfer to other state-of-the-art CNN models?, (f) Does our attack perform zero-shot transfer to a new dataset?, (g) Are certain CNN optimization techniques robust against such attacks?, (h) Do input transformation-based methods provide defense against such attacks?, (i) Can our proposed attack for discriminative models also attack generative models like GANs?, (j) Can our proposed robust feature learning defend against such attacks?, and (k) Can our proposed Spatial Value Function sampling successfully transfer attacks to larger datasets?

We test our algorithm on four datasets: MNIST, Fashion-MNIST, SVHN (cropped 32×32 images), and CIFAR10 images. The perturbed MNIST images for $N_p = 1$ are shown in Figure 4-2(a). The optimal perturbations obtained by our algorithm were used to study the robustness to such attacks using three factors: *external regularization*, *model architecture*, and *optimization technique*. Learning perturbations by evolution involves multiple training rounds in each generation. We use two custom CNN models as underlying models in the evolutionary learning stage: GrayNet (24C3-P-48C3-P-256FC-10S) for MNIST, Fashion-MNIST and ColorNet (32C3-32C3-P-64C3-64C3-P-128C3-128C3-P-512FC-10S) for CIFAR10, SVHN dataset. We use four settings of number of pixel perturbation, $N_p = \{1, 2, 5, 10\}$. We discuss the semantic and nuisance sensitivity in the following section.

2.8.1 Learning Curves and Perturbation Samples

Here, we analyze the performance of our proposed CMA-ES based attack optimization algorithm with increasing generation. We examine test accuracy with increasing generations of our proposed algorithm, as shown in Figure 4-2(c) and Figure 4-2(d) for MNIST and Fashion-MNIST datasets, respectively. For early generations, the pixel attack is sampled from uniformly distributed pixel locations. However, with the CMA-ES optimization, the spatial location, and the intensity of class-wise pixel attacks are optimized, which leads to significant loss of generalization. This is evidenced by the accuracy of test samples, which drops as the optimization advances indicating the soundness of our proposed algorithm. The final learned samples are shown in Figure 3-1 for all the four datasets. Neural networks trained on these attacked datasets show



(a) Partial attack : MNIST

(b) Partial attack: Fashion-MNIST

Figure 2-5: Partially attacking a few images in the training data for (a) MNIST, (b)Fashion-MNIST images. In the x-axis, we show the total number of classes that are attacked. We see that with increasing the number of classes of attack, the testing accuracy almost linearly decreases. This is because out pixel noise selectively attacks samples from a few classes only for the partial attack scenario.

significantly low testing accuracy on clean samples.

GradCAM visualization [90] has been used in several prior works to visualize the spatial distribution of gradients in the input space. Higher CAM values indicate an increased contribution of the input pixel location to the output label. We visualize the mean GradCAM distribution of 100 images per class from the testing dataset corresponding to the true class label for the MNIST dataset for models trained on our proposed attacked dataset in Figure 4-2(b). The CAM distribution shifts its density to non-salient background ROI in the image, thus learning non-discriminative features that do not generalize well. This might explain the drop in testing accuracy with increasing epochs.

2.8.2 Comparison to Prior Methods

We believe our work is the first attempt towards adversarial training time attack using discrete pixel attacks to induce overfitting in neural networks. There are not many previous works on this topic. We choose the work of Jacobsen et al. [48] as our

Table 2.1: Showing testing accuracy (in %) on clean test samples, trained on attacked samples with data augmentation for 30 epochs on the SVHN dataset. Experiments are repeated three times. Our attack method outperforms the previous attack method outlined in [48] due to perturbation optimization using CMA-ES.

Method	**ResNet-20**	**ResNet-32**	**DenseNet-40**
SVHN (clean)	93.5 ± 0.9	92.8 ± 1.0	92.3 ± 1.2
$N_p = 1$ [48]	91.8 ± 0.2	90.9 ± 1.8	91.0 ± 0.4
$N_p = 1$ [ours]	31.3 ± 6.3	37.2 ± 10.4	32.1 ± 9.4
$N_p = 2$ [ours]	14.9 ± 2.4	18.4 ± 3.8	18.8 ± 4.7
$N_p = 5$ [ours]	$\mathbf{9.3 \pm 0.9}$	$\mathbf{11.0 \pm 0.3}$	$\mathbf{16.1 \pm 8.4}$

Table 2.2: Showing testing accuracy (in %) on clean test samples when trained on our proposed attacked samples with data augmentation for 30 epochs. Experiments are repeated three times. Our attacks learned on our custom ColorNet model can transfer to state-of-the-art CNN architecture, causing overfitting with low test and high training accuracy.

Dataset	$\mathbf{N}_p$	**ResNet-20**	**ResNet-32**	**DenseNet-40**
CIFAR10	0	$\mathbf{78.5 \pm 1.2}$	$\mathbf{75.2 \pm 3.0}$	$\mathbf{82.3 \pm 1.5}$
	1	33.3 ± 8.4	30.7 ± 4.2	29.9 ± 2.9
	2	25.5 ± 1.1	$\mathbf{20.7 \pm 6.2}$	23.4 ± 0.7
	5	$\mathbf{14.5 \pm 2.6}$	21.4 ± 1.1	$\mathbf{24.6 \pm 5.2}$
SVHN	0	$\mathbf{93.5 \pm 0.9}$	$\mathbf{92.8 \pm 1.0}$	$\mathbf{92.3 \pm 1.2}$
	1	31.3 ± 6.3	37.2 ± 10.4	32.1 ± 9.4
	2	14.9 ± 2.4	18.4 ± 3.8	18.8 ± 4.7
	5	$\mathbf{9.3 \pm 0.9}$	$\mathbf{11.0 \pm 0.3}$	$\mathbf{16.1 \pm 8.4}$

baseline for prior models that use heuristic pixel placement for attacking the training images. Our method consistently outperforms the baseline method on the metric of test accuracy on the clean test set for all the datasets, as shown in Table 2.1. Our method shows superior performance compared to [48] because we perform optimization to search for the best corruption pattern, whereas the baseline uses a heuristic pixel placement to corrupt the data. These experiments were performed in the presence of random image crop data augmentation.

Qualitatively, our method has similar human imperceptibility as one-pixel evasive attacks like [97] because both methods use one-pixel attacks for fooling CNNs. How-

Table 2.3: Transferring our attack across datasets from CIFAR10 to STL10 dataset. We show that the drop in test accuracy due to our attacks on the source dataset can also be transferred to a target dataset.

	Clean		$N_p = 1$		$N_p = 5$	
	CIFAR	STL	CIFAR	STL	CIFAR	STL
Airplane	75.1	74.5	22.8	26.5	18.3	11.3
Cat	60.5	31.25	16.9	13.1	15.9	7.3
Deer	69.3	61.1	71.8	24.6	4.9	2.6
Dog	56.9	18.9	14.6	8.3	19.7	13.6
Ship	82.0	64.5	7.7	3.5	6.6	13.8
Truck	76.6	52.9	25.8	30.6	41.1	23.3

ever, compared to JSMA [75] which corrupts multiple pixels, our method has better human imperceptibility making it harder to detect. Compared to other training time attacks like [12,48], our method has better imperceptibility because it only attacks a few pixels.

2.8.3 Partially attacking few Classes

We also show the result of training CNNs (GrayNet) with partial class-wise attacks. For example, if we choose the number of classes to attack as k, then training images belonging to classes $\{0, 1, \dots k-1\}$ are corrupted by our proposed pixel-based attack, and other images are kept intact. Figure 2-5 shows the result of testing accuracy on the test data for MNIST and Fashion-MNIST data when the training images are incrementally corrupted for each class. The results show a linear trend of decreasing testing accuracy as more classes are incrementally corrupted, suggesting that our attacks can partially confuse the CNNs for the specific attacked classes while the other classes are correctly classified. Therefore, our proposed attack acts as a mask that hides class-specific features and makes the CNN overfit on the spurious pixel disturbance.

2.8.4 Explicit Regularizations are Easily Overfitted

We are interested in understanding how different factors in neural network learning contribute to robustness against our proposed attacks. Zhang et al. [113] showed that explicit regularization methods have a limited effect in controlling neural networks fitting random noise and labels. In the same spirit, we set up experiments to study the robustness of commonly used regularization techniques against our crafted attacks. Testing accuracy on these methods is shown in Figure 2-6.

Data augmentation: We use random image transformations like cropping, flipping, and zooming, which are used to augment the training data. Data augmentation is the most effective explicit regularization according to our study. This is not surprising because it introduces disturbances to the training data distribution, thus diminishing the effect of the perturbation, which was designed on a fixed dataset with data augmentation. However, we observe that for an increased number of pixel perturbation per image, $N_p \geq 2$, even data augmentation is vulnerable to our attacks.

Dropout [95]: This regularization technique randomly masks layer outputs to reduce the reliance on the output on particular neurons. We used a dropout probability of 0.4. However, dropout seems to have little to no effect on the generalization ability. Since the perturbations are extremely localized in space, we believe that dropout has a negligible effect in consistently masking such spurious artifacts.

Weight decay: This method constrains the norm of the parameters with a Euclidean ball whose radius is determined by the λ co-efficient. It is also known as l_2 regularization or Tikhonov regularization [33]. We use $\lambda = 0.01$. Although weight decay marginally improved the test accuracy at initial epochs, final test accuracy after 30 epochs is similar to that without regularization.

2.8.5 Attack Transfer across Models

Previous works have shown that the choice of model architecture can act as implicit regularization. Statistical machine learning theory predicts that models with a larger number of parameters have higher complexity, making them more likely to converge at a local minimum with poor generalization ability. Li et al. [59] showed that ResNet [42] with skip connections produced a smooth loss surface compared to those without skip connections, hinting that model architecture might play some role in generalization performance. We train state-of-the-art CNN models (with data aug-

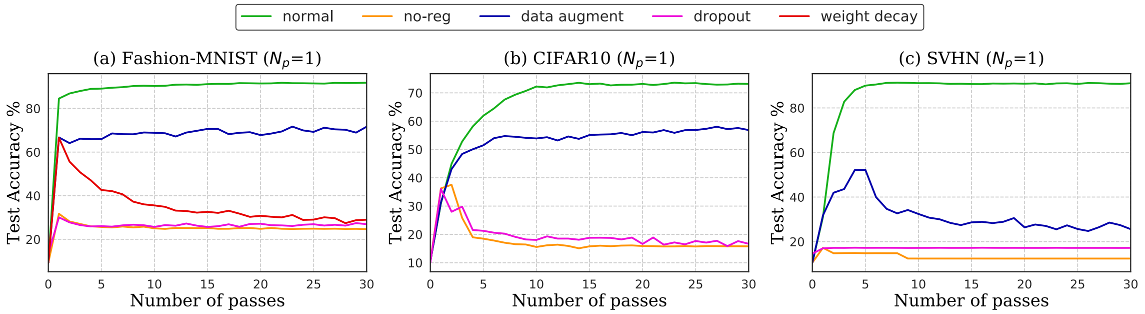


Figure 2-6: Testing accuracy with increasing training epochs for different regularization methods under a single-pixel ($N_p = 1$) attack. *normal* refers to no corruption in training data, and *no-reg* refers to the case where no explicit regularization was used. Experiments were repeated five times and the mean is reported. Training accuracies are close to 100%.

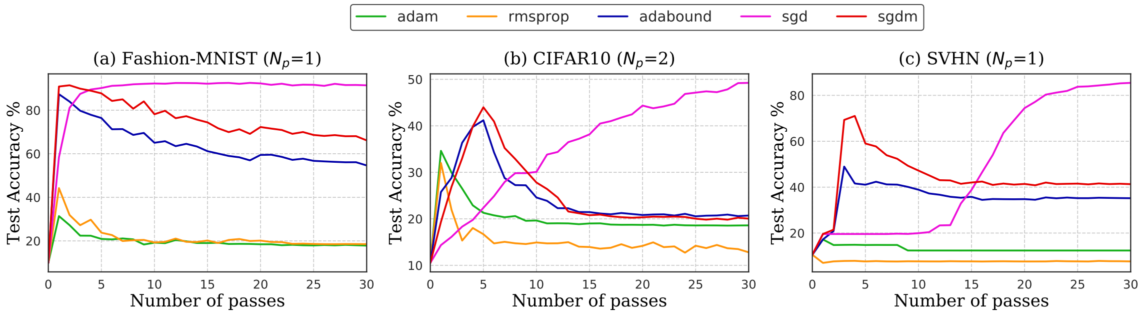
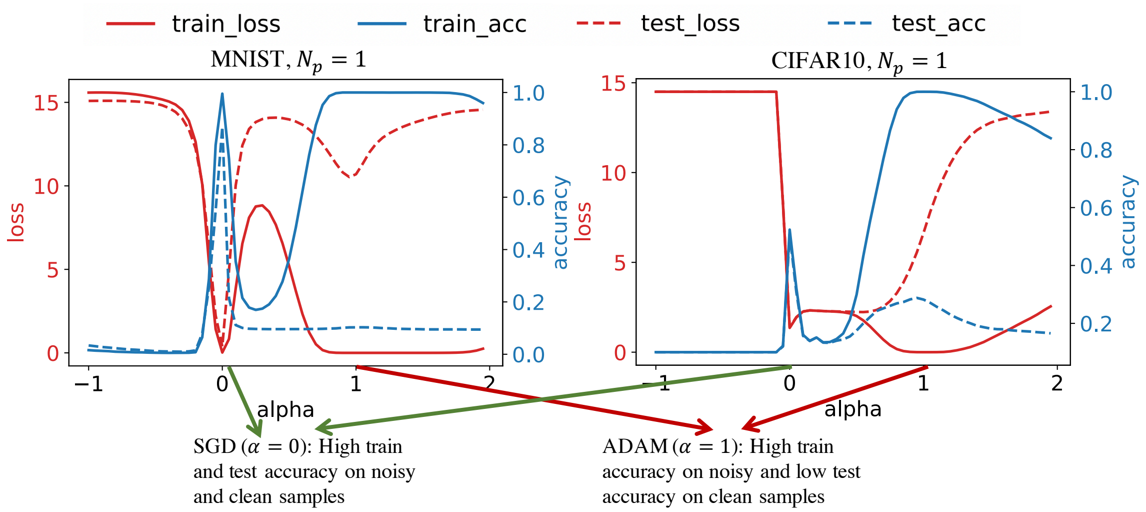


Figure 2-7: Testing accuracy using various optimization strategies under single-pixel perturbation shows SGD consistently performs better than adaptive optimization techniques. Each experiment was performed five times and the mean is reported.

mentation), Resnet-20, Resnet-32 [42], and DenseNet-40 [46] on our attacked training samples learned from our custom-designed CNN models and measure the testing accuracy after 30 epochs, as shown in Table 2.2.

Empirical evaluation reveals a significant difference in test accuracy for unperturbed train images and even a single pixel perturbed data. For different perturbation levels, we do not find a strong correlation between depth and testing accuracy for ResNet models. For example, while ResNet-20 produces better test accuracy on the CIFAR10 dataset than ResNet-32, the opposite is true for SVHN. Therefore, in the presence of such conflicting evidence, it is difficult to convincingly conclude that shallower models are more robust to overfitting than their deeper counterparts.



SGD ($\alpha = 0$): High train and test accuracy on noisy and clean samples

ADAM ($\alpha = 1$): High train accuracy on noisy and low test accuracy on clean samples

Figure 2-8: Plotting loss surface by interpolating from SGD ($\alpha = 0$) to Adam ($\alpha = 1$) weights. The loss surface around the SGD parameter is sharper; however, it has better generalization.

2.8.6 Attack Transfer across Datasets

Similar to testing the transfer of our attack across models, we also show that our proposed attack can be transferred across datasets as well. For this purpose, we trained our CNN model (**ColorNet**) on the source dataset, CIFAR10, with and without the proposed attack. The same model is then tested on the STL10 dataset [?] which has similar labels to the CIFAR10 datasets. We choose the common labels between two datasets, {`airplane, cat, deer, dog, ship, truck`}, for reporting the test accuracy. Table 2.3 shows the accuracy on the test set for both the datasets. For clean images, images belonging to the same labels show similar accuracy. However, when trained on our proposed attacked images, the testing accuracy shows a drop for both the source and target datasets. Although the attack pixels were not trained for the STL10 dataset, our attack is shown to apply to other datasets also in a zero-shot manner.

2.8.7 Adaptivity can Overfit to Proposed Attacks

High out-of-sample error is generally attributed to poor convergence of the neural network parameters to an unfavorable local minimum. By examining the robustness

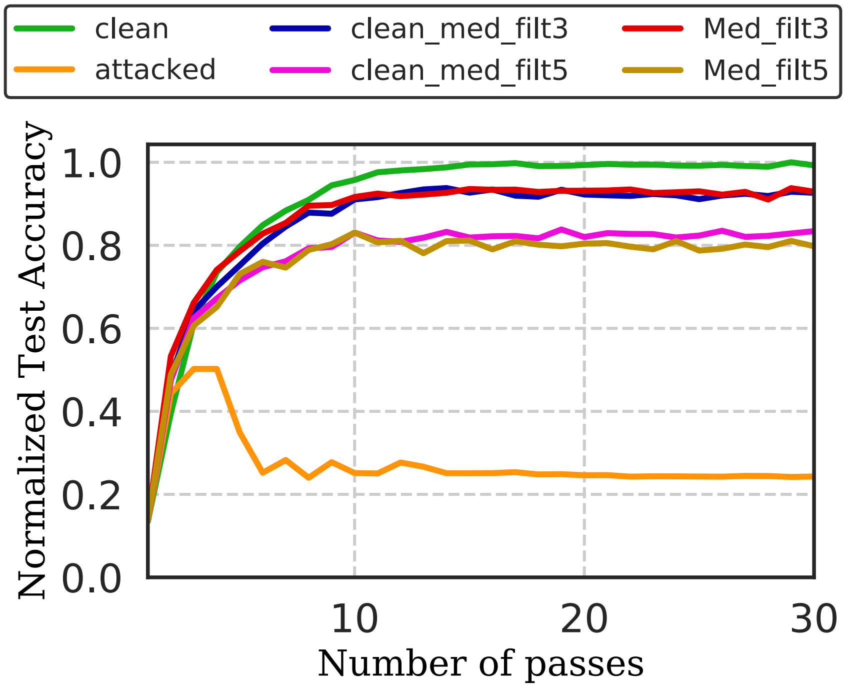


Figure 2-9: Median filtering based defense against our proposed pixel-based noise for CIFAR10 dataset. Median filtering improves the performance due to the removal of pixel-noise. However, it also brings down the performance of training on clean images due to the removal of certain high-frequency features due to the filtering process.

of well-known optimization strategies to our pixel-wise attacks, we wish to study if a certain algorithm is more liable to memorizing small perturbations while ignoring other salient statistical patterns in the training data. To this end, we trained CNN models on single-pixel perturbed data using Adam [53], SGD, RMSProp [101], and Adabound [62] optimization. The results are shown in Figure 2-7.

Wilson et al. [105] showed that adaptive methods are affected by spurious features that do not contribute to out-of-sample generalization by crafting a smart artificial linear regression example. Our method can be viewed as a generalization of such methods for the automatic creation of such spurious examples that scale to arbitrarily sized datasets by gradient-free evolutionary strategies. Figure 2-7 reveals that Adam and RMSProp show prohibitively low testing accuracy for all cases while vanilla SGD is surprisingly resilient to such perturbations showing better out-of-sample performance consistently for all the datasets. Adabound uses strategies from both SGD and Adam, thus showing intermediate performance. It can be concluded that adap-

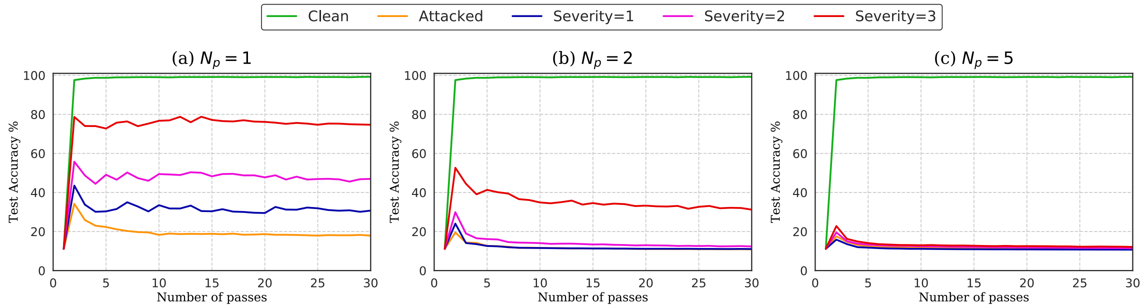


Figure 2-10: Showing the effect of adding Gaussian noise [43] on the attacked images using our proposed pixel-based attack for MNIST dataset. With the increasing severity of the additive Gaussian noise, the defensive properties against our pixel attack are improved. However, for more strength of the pixel attack, additive Gaussian noise is incapable of providing suitable defense.

Table 2.4: Comparison of various learning objectives based on proposed performance metrics. Our proposed loss function (vibCE) has significantly higher semantic feature sensitivity and lower nuisance feature sensitivity (which is desirable for robust classifiers) compared to CE loss due to better semantic feature preservation.

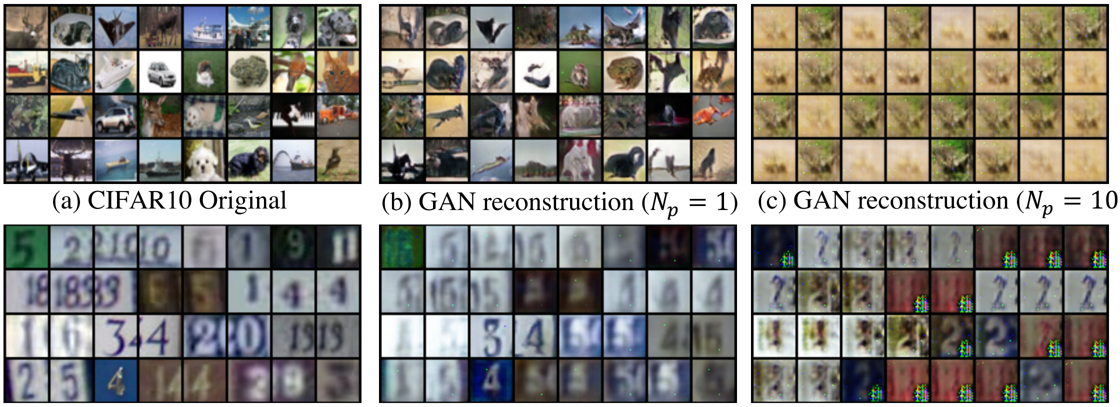
		Semantic Feature Sensitivity (α_S) ↑				**Nuisance Feature Sensitivity (α_N) ↓**			
		MNIST		F-MNIST		MNIST		F-MNIST	
		Random	EvoShift	Random	EvoShift	Random	EvoShift	Random	EvoShift
	CE	0.99		0.9		0.02		0.01	
Clean	vibCE (ours)	0.98		0.84		0.02		0.01	
	CE	0.30	0.01	0.32	0.16	0.92	0.93	0.94	0.89
$N_p = 1$	vibCE (ours)	**0.95**	**0.94**	**0.72**	**0.73**	**0.01**	**0.01**	**0.09**	**0.09**
	CE	0.32	0.01	0.30	0.13	0.99	1.00	0.99	0.72
$N_p = 2$	vibCE (ours)	**0.94**	**0.94**	**0.75**	**0.64**	**0.03**	**0.02**	**0.09**	**0.11**
	CE	0.15	0.00	0.27	0.12	1.00	1.00	0.97	0.99
$N_p = 5$	vibCE	**0.93**	**0.90**	**0.70**	**0.63**	**0.09**	**0.07**	**0.20**	**0.27**

tive methods heavily overfit to training input perturbations while vanilla SGD is considerably robust to such changes.

Loss surface: Keskar et al. [45, 51] claimed that flatter minima solutions generalize better than their sharper counterparts. To investigate this phenomenon, we visualize the loss surface around the learned parameters by interpolating the weights obtained from SGD and Adam optimization following the strategy by Goodfellow et al. [36]. We plot the loss function values and train/test accuracies at intermediate intervals given as

$$\boldsymbol{w}_\alpha = \alpha \boldsymbol{w}_{\text{Adam}} + (1 - \alpha)\boldsymbol{w}_{\text{SGD}}, \tag{2.14}$$

as shown in Figure 2-8. Interestingly, we find that SGD finds sharper minima solutions



(a) CIFAR10 Original (b) GAN reconstruction ($N_p = 1$) (c) GAN reconstruction ($N_p = 10$)

(d) SVHN Original (e) GAN reconstruction ($N_p = 1$) (f) GAN reconstruction ($N_p = 10$)

Figure 2-11: Generated samples by GANs on attacked data distribution show that semantic features in the true samples are suppressed by our proposed training attacks resulting in poor reconstruction of images. The quality of reconstructed images degrades with increasing attack strength. Spurious features are, however, faithfully reconstructed, indicating over-reliance on such artifacts by the discriminator.

where both test and train loss are lower ($\alpha = 0$) than Adam, whereas the training loss exhibits are flatter geometry ($\alpha = 1$). This pattern is repeatedly visible for all datasets suggesting that sharpness of minima does not guarantee a solution that has better generalization robustness to training perturbations, which is along the same line of argument as claimed by Dinh et al. [25].

2.8.8 Effect of Input transformations

Previous defensive methods in adversarial defense [109] show that input transformation based defense are effective against a certain class of adversarial attacks. Therefore, we compare our proposed method against two kinds of input transformations: (i) median filtering, and (ii) additive Gaussian noise of varying severity.

Median filtering: Since median filtering is well-known to prevent salt-and-pepper noise, we applied median filtering on our attacked images to find the resilience of our method against such a defensive strategy. Figure 2-9 shows the effect of median filtering on the CIFAR10 dataset for the number of pixel attack, $N_p = 1$ as measured by normalized test accuracy with respect to the clean training data accuracy. In the presence of median filtering, we observe improvement in the testing accuracy when

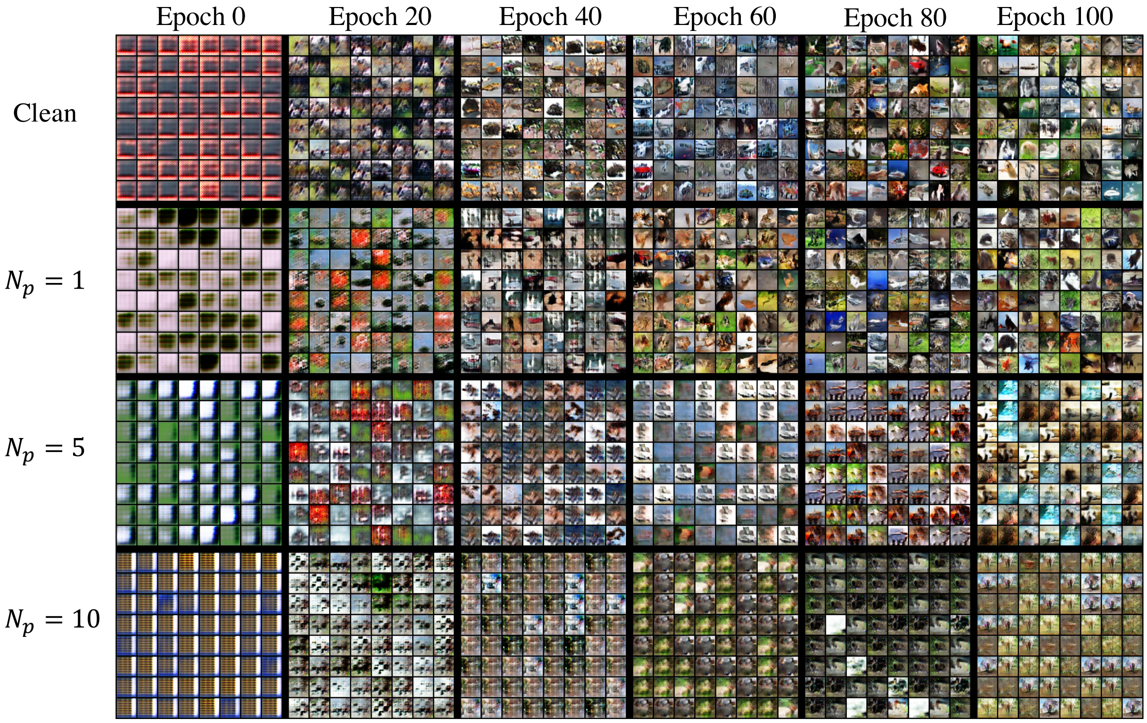
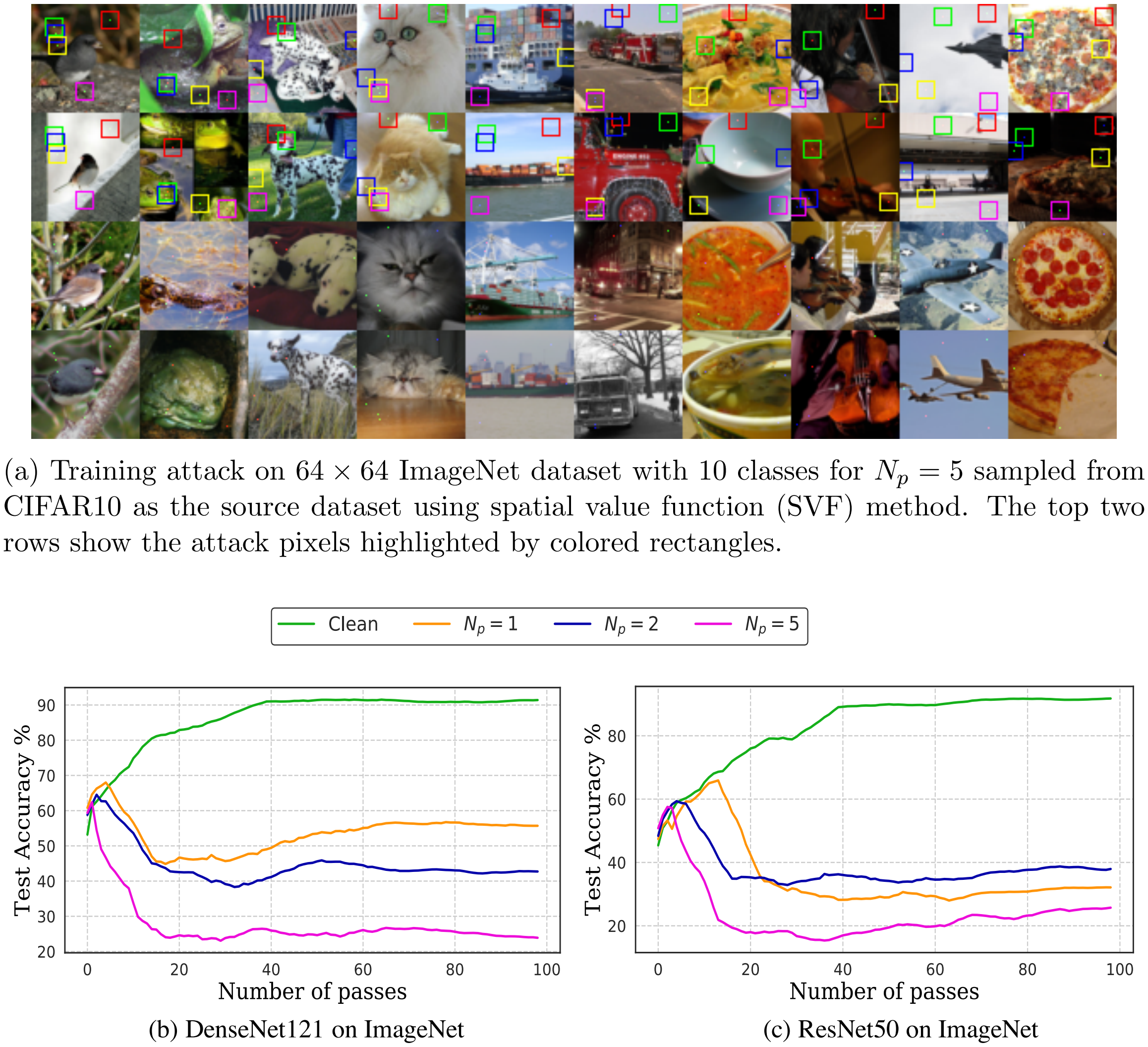


Figure 2-12: Showing the progression of GAN training in the presence of our proposed training-time attack data using VAEGAN [57]. For the 'clean' data, the generated images resemble natural images with the progression of training. However, in the presence of our proposed pixel attack, the generated images do not generate natural features like object shape and color but enhance the attacked pixels (Best viewed in color and zoomed in).

compared to the attacked training scenario. However, there is a slight drop in the clean testing accuracy as well when median filtering is applied. This is due to the removal of certain detailed features in the image due to the filtering process.

Although median filtering improves the testing accuracy in the presence of our proposed training-time attack, this is also the case for well-known adversarial attacks [16,35,63,98] that can be defended against by simple input transformation-based methods proposed in [38, 109]. Therefore, similar to previous methods in adversarial attacks, we believe this does not undermine the contribution of this method, which proposes a novel method for attacking training images, thus uncovering a new kind of vulnerability in neural network learning.

Additive Gaussian noise: We wanted to test the effect of adding random noise to our proposed pixel-based attack on training images. To that effect, we added



(a) Training attack on 64 × 64 ImageNet dataset with 10 classes for $N_p = 5$ sampled from CIFAR10 as the source dataset using spatial value function (SVF) method. The top two rows show the attack pixels highlighted by colored rectangles.

(b) DenseNet121 on ImageNet

(c) ResNet50 on ImageNet

Figure 2-13: Degrading discriminative performance on ImageNet 64×64 dataset with increasing pixel perturbation strength.

Gaussian noise from [43] with various severity on the attacked images. Details on the severity levels for the Gaussian noise is explained in [43]. Figure 2-10 shows the effect of testing accuracy when the CNN model is trained in the presence of both our proposed pixel attack and random Gaussian noise of various severity. The results show that for a few pixel attacks, high severity Gaussian noise can provide defense against our attacks. We hypothesize this is due to masking of the pixel attacks by the additive noise which diffuses its strength. However, with more number of pixel attacks, additive Gaussian noise does not provide much defense as shown by the low test accuracy for $N_p = 5$ case. Therefore, our proposed attack (with a high number of pixels) is resilient against additive random noise.

2.8.9 Poor Generative Adversarial Network reconstruction under EvoShift

Since GANs use cross-entropy loss in the discriminator for classification between generated images and images from data distribution, our proposed training time attacks can confuse the discriminator. To study this effect, we sample images from our proposed EvoShift-ed version of standard datasets, $x \sim \mathcal{D}_{adv}$, and learn VAEGAN [57] on such images. We show qualitative comparisons of the GAN reconstruction under our proposed attack in Figure 2-11. We find that spurious few-pixel perturbations can effectively mask the true data distribution, resulting in large degradation of reconstructed images from the GAN. With the increasing strength of the number of pixels in the attack, the quality of reconstruction increasingly degrades, indicating high nuisance sensitivity of GAN discriminators.

Corresponding to the reconstructions in Figure 2-11, for CIFAR10 images, we obtains a PSNR of 14.42 dB for $N_p = 1$ and 11.06 dB for $N_p = 10$. For the SVHN dataset, these values are 16.20 dB and 13.49 dB respectively. Quantitative analysis demonstrates poor reconstructed image qualities in terms of low Peak Signal to Noise ratio (PSNR) by VAEGAN under our proposed EvoShift. This implies that with the increasing strength of the attack, GANs ignore semantic features in the images and confuse the spurious artifacts as true data distribution, which is an undesirable vulnerability in generative models that have not been studied by previous works.

Figure 2-12 shows the progression of GAN training in the presence of our training-time pixel-based attacks for the CIFAR10 dataset. For clean images, the GAN learns to progressively learns to generate semantic features that are related to natural images. On the other hand, in the presence of our training-time attacks, the GAN model cannot generate the semantic features but focuses on generating the noisy pixel features. With the increasing strength of the attack, as measured by the number of pixels, the generated images show increased natural feature suppression with an increased focus on generating the noisy pixels. The discriminator can be attacked by our proposed pixel-based method that overfits the noisy pixel features.

2.8.10 Robustness of Our Variational Objective

Table 2.4 provides the performance comparison in terms of the proposed metrics for different loss functions: CE and our proposed vibCE corresponding to the MNIST and Fashion-MNIST dataset, respectively. *Random* refers to a uniform spatial sampling of pixel perturbation as the attack. This is the case where no training-time attack optimization has been performed using the evolutionary strategy and corresponds to the initial solution of the optimization. We infer two major insights from the results: (1) Our proposed EvoShift outperforms the random attack sampling case shown by lower α_S and high α_N for both CE and vibCE loss. This shows that our proposed EvoShift algorithm finds suitable pixel attack parameters that overfit the model to training data that is not possible by attacking with random pixel placement, (2) Our robust objective demonstrates significantly higher semantic sensitivity (α_S) and low nuisance sensitivity (α_N) compared to CE loss. Training with vibCE loss function retains the semantic features related to shape and color information and thus does not overfit to the additive adversarial pixel attacks during training.

2.8.11 Scaling attack to ImageNet dataset

In this work, we are trying to show vulnerabilities in neural networks by attacking training data (not test data) which requires multiple training on the perturbed dataset. Due to computational expenses, this is difficult to achieve. However, here we show that using the SVF sampling method, we can successfully scale such attacks from CIFAR10 to ImageNet samples. Figure 2-13(a) shows few samples from our training pixel perturbations for 64×64 ImageNet [22] dataset using our proposed Spatial Value Function (SVF) based transfer from CIFAR10 dataset which has an image shape of 32×32. Figure 2-13(b)-2-13(e) shows degradation in classification performance under attacked training images for ResNet-50, ResNet-101 [42], and DenseNet-161 [46] models. We see that even a single pixel attack on the training images can bring down the testing accuracy on clean images to almost 50%. Increasing the attack severity to $N_p = 5$ and $N_p = 10$ can further degrade the testing accuracy. Thus, our proposed SVF methods for transferring attacks from source to target dataset is effective and can reduce the testing accuracy, without the need for recomputing the pixel optimization by CMA-ES. However, performing CMA-ES in addition to SVF based transfer might even strengthen the attack further.

2.9 Conclusions

We present an adversarial training time attack using a population-based evolutionary strategy along with a novel fitness score designed to explicitly maximize domain divergence and generalization gap. We observe that it is possible to fool neural networks with each passing generation suggesting that specific spatial locations exist on the input image that are more vulnerable to being attacked than others. This result exposes serious vulnerabilities in CNNs. Our analysis reveals that a proper selection of the optimization technique is paramount to good generalization properties. We found that SGD performs significantly better than adaptive optimization methods in ignoring spurious training features that do not contribute to the out-of-sample generalization. Our analysis of loss surface reveals that SGD finds sharper minima solutions despite good generalization performance. Such training distribution attacks can also be extended to GAN discriminators causing poor reconstruction of semantic components in the image. This work is one of the first works in the field of attacking GANs using spurious adversarial noise in training data. Furthermore, we showed that this vulnerability in neural networks is related to the inefficiency of cross-entropy loss. We also proposed a robust loss function based on variational inference principles that increase the mutual information between semantic features and the labels resulting in improved performance measured by the sensitivity measures. In this work, we have provided an extensive analysis of the behavior of CNNs in the presence of intelligently crafted adversarial training noise. We believe this work will fuel further research into understanding the robustness of deep learning algorithms regarding generalization in the presence of training time adversarial attacks.

Chapter 3

Robustness of Neural Network Optimization under Training Perturbations

Adaptive gradient methods such as Adam, RMSProp, AdaGrad use the temporal history of the gradient updates to improve the speed of convergence and reduce reliance on manual learning rate tuning, making them a popular choice for *off-the-shelf* DNN optimizers. In this work, we study the robustness of neural network optimizers in the presence of training perturbations. We show that popular adaptive optimization methods exhibit poor generalization to clean test data, compared to vanilla Stochastic Gradient Descent (SGD) and its variants, which manifest better implicit regularization properties. We construct an illustrative example of a family of two-class linearly separable toy-data such that models trained under noise using adaptive optimizers show only 52% test accuracy (random classifier). Stochastic gradient methods can achieve 100% test accuracy. We strengthen our hypothesis by empirical analysis using CNNs on publicly available image datasets, which further highlights the robustness of SGD optimization against such noisy training data compared to its adaptive counterparts. Based on the results, our work suggests a reconsideration of the extensive use of adaptive gradient methods for neural network optimization, especially when the training data is noisy.

3.1 Introduction

Deep Neural Networks [58] are high capacity models where the number of learnable parameters is often significantly more than the number of training samples. In such an over-parametrized setting with highly non-convex loss surface, classical learning theory [8, 103] predicts a high out-of-sample error because the solution is likely to get stuck at a local minimum. Nonetheless, deep neural networks appear to generalize well even in small data regimes [72] and have shown state-of-the-art performance in many practical tasks. Optimization methods for DNNs play an important role in finding network parameters that converge fast and generalize well to unseen data samples from the underlying distribution.

There have been numerous works in the field of DNN optimization starting from vanilla Stochastic Gradient Descent (SGD) to sophisticated adaptive gradient methods. Given a cost function $J(\theta)$ parametrized by the weights of the neural network θ, a standard SGD update rule finds the direction of descent and updates the parameters in that direction given as

$$\theta = \theta - \eta \cdot \nabla_\theta J(\theta). \tag{3.1}$$

Over the years, many variants of SGD have been proposed that improve the acceleration and convergence properties of the optimization. Momentum-based methods [69, 81] determine the direction of movement towards the local minima akin to a ball rolling in a curved surface. Recently, gradient-based methods [27, 53, 112] adaptively update the learning rate to enable accelerated convergence without learning rate tuning efforts. Hence, these methods are popular *off-the-shelf* choice for DNN optimization.

Motivation: While adaptive methods perform well on various tasks without the need for manual learning rate tuning, adaptivity can have its dangers. In this work, we benchmark and compare the performance of adaptive and non-adaptive gradient methods in the presence of training perturbations. Training in the presence of noise is a real problem that can occurs in noisy acquisition devices. Therefore, benchmarking the robustness of optimizers against such training noise is an important task, which gives us a measure of whether the features selected by the optimizer conform to semantic information like color and shape (for images) that enable better generalization to non-noisy test samples.

Contributions: In this work, we train DNN models on noisy training data using various optimizers and measure the performance of such models on clean test data, thus benchmarking how liable the optimizers are to overfit the training noise. Our first contribution is the construction of a linearly separable two-class toy dataset upon which we superimpose a crafted noisy signal. Following that, we analytically show that adaptive gradient methods completely fail to learn any patterns from the data and do not generalize to the clean test set. On the other hand, SGD and its variants show 100% test accuracy on the test-set showing greater robustness against such spurious noise.

Secondly, for higher-dimensional image datasets, we use a gradient-free noise optimization, based on [18] for finding optimal pixel perturbations that maximize the generalization gap between training and testing images. CNN models are trained on such worst-case noisy images using various optimizers. These trained models are then evaluated on clean data to measure the generalization of optimization methods. Empirical studies on MNIST, CIFAR10, and SVHN dataset confirm our hypothesis that vanilla SGD and its variants are significantly more robust against such perturbations compared to adaptive gradient methods. Our analysis of the 2D loss surface reveals that SGD tends to find solutions around flatter loss regions, which might explain our empirical observations. Based on our benchmarking results, we recommend using SGD optimizers with learning rate tuning instead of adaptive gradient methods, especially when there exists some training noise or distribution shift between the training and validation/testing data.

3.2 Related Works

We present some previous works on generalization in neural networks and its effect based on optimization strategies.

3.2.1 Generalization in DNNs

Since DNNs usually have more number of parameters compared to the number of training samples, there is a high chance of overfitting according to classical learning theory. There has been some work [41, 70, 78, 106] that study the generalization properties of neural networks under such high complex over-parametrized settings, whereas classical learning theory suggests difficulty in training due to high variance solutions. Various recent works have sought to explain generalization in neural networks. [113] showed that neural networks can fit random noise. The idea of pixel perturbation has also been explored in [114] to measure the testing accuracy of images.

3.2.2 Generalization of SGD optimization

Previous works have analyzed the trade-offs between vanilla SGD and adaptive methods. Some works [13, 14, 27, 41] argue that SGD results in solutions that generalize well to the test set. However, [106] claimed that SGD is not the key to generalization and showed that other optimization methods can generalize with similar performance. Another approach [45, 51] to answer why neural networks generalize well, studies the loss surface geometry around the learned parameter, and shows that sharper minima solutions tend to generalize poorly compared to flatter minima which were contested by [25].

Some recent research [52, 62, 84, 105] also demonstrate that vanilla SGD optimization has better generalization ability than adaptive optimization methods. [71] discuss implicit regularization induced by various algorithmic choices in deep learning model selection. Although there has been work in this area, a consensus does not exist concerning why neural networks exhibit such well-behaved generalization properties and how different optimizations contribute to the generalization of such over-parametrized models. Unlike previous works, the novelty of our method lies in analyzing the robustness of neural network optimizations under training perturbations, which has not been studied in detail by previous works.

3.3 Neural Network Optimization

A good survey of the various optimization methods used in deep learning is provided by [88]. Given $J(\theta)$ as the cost function of the neural network with θ as the parameter to optimize, the following is the description of the update rule for popular DNN optimizers.

Stochastic Gradient Descent (SGD): SGD is the simplest method that computes the gradient of the cost function J w.r.t. to θ for small batches of dataset. This is also known as mini-batch gradient descent, with update equation given by Equation 3.1, where ∇ is the learning rate that decides the rate at which the parameter moves toward the local minima.

Nesterov Accelerated Gradient (NAG): SGD based optimization has difficulty in regions of the loss surface having high gradients in a particular direction, where the updates oscillate around the slope of the dominant direction making small progress in the direction of the local minimum. Momentum based methods [81] take into account the relevant direction of motion based on the previous update, thus accelerating the update toward the local optimum. This is similar to a ball rolling in a curved surface gaining momentum due to gravity. Nesterov accelerated gradient (NAG) [69] is a modification of the momentum-based update which uses a look-ahead step to improve the momentum term, given as

$$\begin{aligned} v_t &= \gamma\, v_{t-1} + \eta \nabla_\theta J(\theta_t - \gamma v_{t-1}), \\ \theta_{t+1} &= \theta_t - v_t, \end{aligned} \tag{3.2}$$

where γ is the momentum term usually set around 0.9.

RMSprop: RMSProp [100] attempts to automatically tune the learning rate by diminishing it with the root mean square of the accumulated gradients from the previous updates. The update equation is given as

$$\begin{aligned} E[g^2]_t &= 0.9E[g^2]_{t-1} + 0.1g_t^2, \\ \theta_{t+1} &= \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} g_t. \end{aligned} \tag{3.3}$$

Gradually the learning rate is diminished to a small value allowing updates to be automatically scaled near the local minima.

Adam: In addition to the mean of the past gradients, Adam [53] also stores the uncentered variance of the past gradients,

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1-\beta_1) g_t, \\ v_t &= \beta_2 v_{t-1} + (1-\beta_2) g_t^2. \end{aligned} \tag{3.4}$$

The author also performs a bias correction for the above parameters which are initialized to zero. The update equation is given as

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t. \tag{3.5}$$

Default values of 0.9 for β_1 and 0.999 for β_2 are commonly used. Adam is a popular choice of optimization among deep learning practitioners because it enables accelerated convergence in most settings without manual learning rate tuning.

3.4 Common Framework for Optimization

Gradient updates for the above methods were unified in a common framework by [105], where they expressed the update equation as follows,

$$\begin{aligned} \theta_{t+1} = \theta_t &- \alpha_t \mathrm{H}_t^{-1} \tilde{\nabla} f(\theta_t + \gamma_t(\theta_t - \theta_{t-1})) \\ &+ \beta_t \mathrm{H}_t^{-1} \mathrm{H}_{t-1} (\theta_t - \theta_{t-1}). \end{aligned} \tag{3.6}$$

The matrix $\mathrm{H}_t := H(\theta_1, \cdots, \theta_t)$ is diagonal which is usually defined as

$$\mathrm{H}_t = \mathrm{diag}\left(\left\{ \sum_{i=1}^{t} \eta_i g_i \circ g_i \right\}^{1/2} \right), \tag{3.7}$$

where $g_t = \tilde{\nabla} f(w_t + \gamma_k (w_t - w_{t-1}))$ is the gradient signal at time step t used for the update. Therefore, H_t is a diagonal matrix that stores the weighted cumulative sum of the squares of past gradients for adaptive gradient methods. For SGD and its variants, $\mathrm{H}_t = I$ because they do not use the cumulative sum of squares of previous gradients in the current update.

Using the above formulation, [105] states that SGD and its variants will converge to the minimum ℓ_2 norm solution. They develop the following lemma for adaptive gradient methods, which we will use to find a family of worst-case perturbations for a linearly separable toy dataset that adaptive gradient methods are not robust against.

For simplicity, let us consider the binary least-squares classification set up with the objective,

$$J(\theta) = \frac{1}{2}\|X\theta - y\|_2^2. \tag{3.8}$$

Lemma 3.4.1. *Starting from the initial weight of θ_0=0, if there exists a solution for θ in Equation 3.8 that lies in the direction of* $\mathrm{sign}(X^T y)$*, i.e.* $X\ \mathrm{sign}(X^T y) = cy$ *for some scalar c, then the final solution for the neural network parameters follow* $\theta \propto \mathrm{sign}(X^T y)$*, where* $\mathrm{sign}(x)$ *maps each component of x to its sign.*

The basic outline of the proof utilizes the fact that adaptive gradient methods have H_t^{-1} common in all terms of its update Equation 3.6 starting from a zero solution. This common term in all past gradient updates can yield a similar sign to the final learned weights by factoring out the constant terms. SGD based methods use $\mathrm{H}_t = I$ and therefore the above result does not apply to these methods. We direct the readers to [105] for the detailed proof of the above lemma. Using this lemma, we put forward a class of perturbations in a toy dataset and prove that adaptive gradient methods trained in the presence of such perturbation are incapable of generalizing to the test set in the presence of our crafted perturbations, although being linearly separable.

3.5 Adaptive Gradient Methods Fail on Linearly Separable Toy-Data

Our goal is to benchmark the robustness of optimization methods in the presence of spurious training perturbations. Let us define a classifier $h : X \to Y$ from a hypothesis space $\mathcal{H}$. The true data distribution containing the semantics of the class is given as $(\boldsymbol{x}_i^s, y_i^s) \sim \mathcal{D}$ from which we sample the clean samples. Let us consider noise sampled from $\boldsymbol{x}_i^n \sim \mathcal{D}_\eta$ which is then added to the true samples to obtain training samples $\boldsymbol{x}_i = \boldsymbol{x}_i^s + \boldsymbol{x}s_i^n, y_i = y_i^s$. The classifier h is then trained on the training samples $\{(\boldsymbol{x}_i, y_i)\}_i$ and we measure the classification accuracy on test samples $\{(\boldsymbol{x}_i^{test}, y_i^{test})\}_i$ drawn from the true distribution $\mathcal{D}$ defined as $A_{\mathcal{D}}(h) \overset{\text{def}}{=} \mathbb{E}_{(\boldsymbol{x}^{test}, y^{test}) \sim \mathcal{D}}\big(I[h(\boldsymbol{x}^{test}) = y^{test}]\big)$.

Now we build a class of linearly separable datasets such that models trained on superimposed spurious perturbations make adaptive and non-adaptive optimizers behave differently in terms of generalization behavior. Let us consider a dataset (x_i, y_i) of N samples with binary labels $y \in \{+1, -1\}$ with $p > 1/2$ being the probability of positive labels. The feature dimension of x is $m(2N+1) + (N+1)$, where m is a positive constant. We consider original data distribution as

$$x_{ij}^s = \begin{cases} y_i & \text{if } j = 1 \\ 1 & \text{if } j = 2, \ldots, (m+1) \\ 0 & \text{otherwise.} \end{cases} \tag{3.9}$$

The above dataset is linearly separable in the first dimension, and it is trivial to find neural network parameters that correctly classify these samples irrespective of the optimizer used. We add a perturbation on the above dataset of the form,

$$x_{ij}^n = \begin{cases} 0 & \text{if } j = 1, 2, \ldots, (m+1) \\ 1 & \text{if } j = \text{idx}_i^s, \ldots, \text{idx}_i^e \\ 0 & \text{otherwise.} \end{cases} \tag{3.10}$$

where $\text{idx}_i^s = 2 + m + (2m+1)(i-1)$ and $\text{idx}_i^e = 2 + m + (2m+1)(i-1) + m(1-y_i)$ denote the starting and ending index of the i^{th} block of feature. This features takes a moving block of $(2m+1)$ length displaced by $(i-1)$ times and fills it with a single 1 at $2 + m + (2m+1)(i-1)$ position if $y_i = 1$, else fills the entire following $(2m+1)$ positions with 1s.

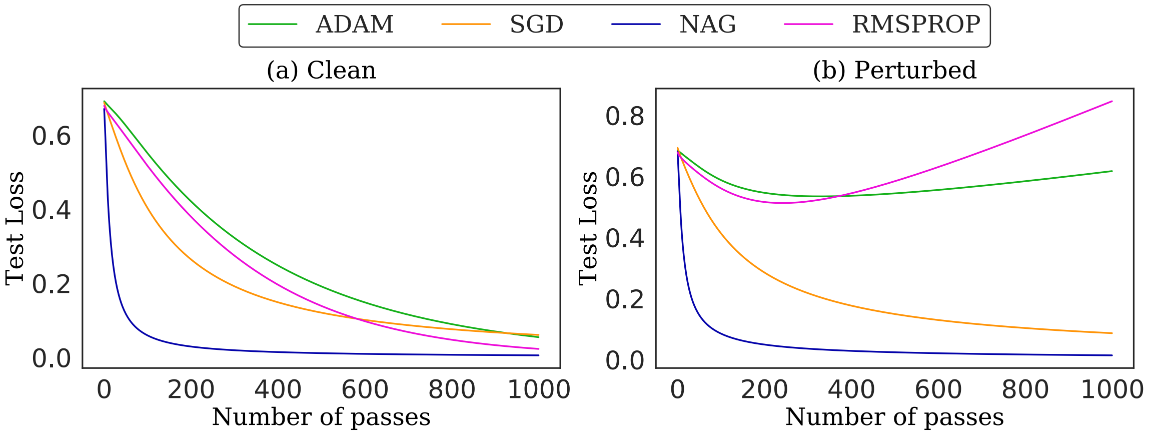


Figure 3-1: Empirical analysis on the toy dataset, (a) The model trained on true samples (clean) shows diminishing test loss for all optimizers due to good generalization to the test set. (b) Test loss on perturbed training data for Adam and RMSProp are divergent showing that they do not generalize to test samples while SGD based methods show a very low test loss. We report the mean of five runs.

The perturbed training data, $\boldsymbol{x}_i = \boldsymbol{x}_i^s + \boldsymbol{x}_i^n$, is then formed by superimposing the nuisance on the true distribution to obtain the training features,

$$x_{ij}^{train} = \begin{cases} y_i & \text{if } j = 1 \\ 1 & \text{if } j = 2, \ldots, (m+1) \\ 1 & \text{if } j = \text{idx}_i^s, \ldots, \text{idx}_i^e \\ 0 & \text{otherwise,} \end{cases} \tag{3.11}$$

with variables having similar meaning as above. It is to be noted that the above-perturbed training data distribution is also linearly separable in the first dimension. Thus it is possible to learn a decision function that uses the first dimension as a discriminative feature that will result in good generalization to the clean test data. However, using Lemma 3.4.1, we show that adaptive methods cannot learn such a decision function. Instead, they learn to predict all test data as positive samples.

Let $v = X^T y$, where X is a matrix of training features and y is a vector of training labels (dropping suffix). Then the component-wise sign of the vector is given as

$$\operatorname{sign}(v_j) = \begin{cases} \operatorname{sign}(\sum y_i^2) = 1 & \text{if } j = 1 \\ \operatorname{sign}(\sum y_i) = 1 & \text{if } j = 2, \dots, m \\ \operatorname{sign}(y_j) = y_j, & \text{if } j > (2+m) \text{ and } x_{c_j,j} = 1 \\ 0 & \text{otherwise,} \end{cases}$$

where $c_j = \lfloor \frac{j+m-1}{2m+1} \rfloor$. Finally, we need to compute the value of X sign$(X^T y)$. The i^{th} entry of X sign$(X^T y) = x_i^T v$, which can be extended as

$$x_i^T u = \underbrace{y_i + m}_{\substack{\text{Due to} \\ \text{true features}}} + \underbrace{y_i(m + 1 - my_i)}_{\substack{\text{Due to} \\ \text{nuisance}}} = (m+2)y_i.$$

Therefore, according to Lemma 3.4.1, adaptive gradient methods will have weights proportional to sign$(X^T y)$. Since the test data x_i^{test} is drawn from the true distribution, only the first $1 + m$ terms are non-zero (Equation 3.9). Therefore, we need to consider the sign of only the first $1 + m$ entries of the weight θ^{adaptive}, which are positive according to sign$(X^T y)$. Assuming a positive constant weight of θ_+, we obtain a decision function as

$$\theta^{\text{adaptive}^T} x_i^{\text{test}} = \theta_+(y_i^{test} + m)\,.$$

Therefore for all values of $m \geq 2$, adaptive gradient methods will always predict positive samples even though it was trained on linearly separable training data.

On the other hand, SGD based methods will find the minimum ℓ_2 normed solution (based on the psuedo inverse) , thus learning the dependence of the first feature dimension leading to successful generalization to the test data even when trained on perturbed data. We validate this point by empirical experiments on the above toy data in the following section.

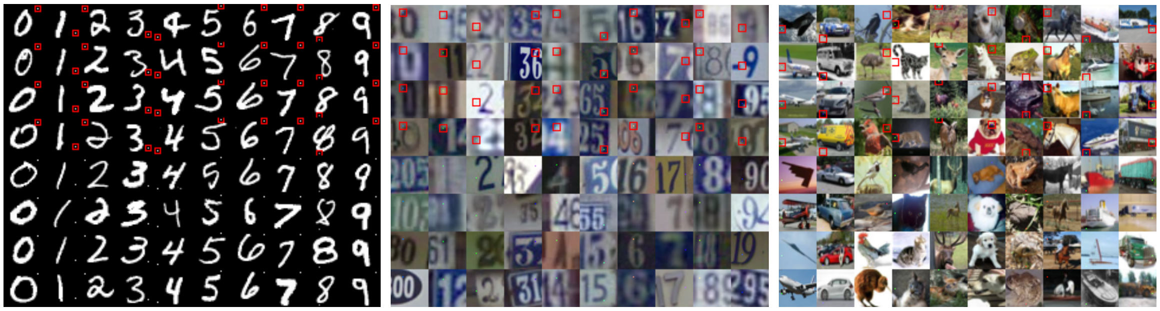
Empirical Analysis: We also performed empirical analysis on a single-layered neural network model to classify the above toy-data into two classes. We choose

Table 3.1: Testing accuracy metrics for vanilla SGD and adaptive optimizers on our crafted toy-dataset. SGD-based methods show 100% robustness to perturbed train data by classifying all test samples correctly. Adaptive gradient methods produce random classification. We report mean of five runs.

Test accuracy	Clean	Perturbed
Adam	100%	52%
RMSProp	100%	52%
SGD	100%	100%
NAG	100%	100%

$N = 100$ samples and $m = 2$, yielding feature dimension of 503. First, we trained four optimizers, Adam, RMSProp, SGD, and NAG, on the clean data defined as Equation 3.9 and tested on clean test samples. Next, we train on perturbed training samples crafted by us following Equation 3.11 using the same four optimizers. For both cases, we perform inference on the same clean test samples. Note that perturbed training samples are also linearly separable on the first dimension as are the clean test samples.

Table 3.1 shows the test accuracy of the optimizers on the toy dataset. Vanilla SGD-based methods perfectly classify the test data for both clean training and the perturbed training case. On the other hand, adaptive gradient methods fail to generalize to clean test data for the perturbed training case as they predict all test data as positive samples. Figure 3-1 shows the behavior of the test loss in the case of a clean and perturbed training case. We observe a divergent test loss curve for adaptive methods signifying that the learned features by such optimizers overfit to the training samples, thus showcasing the lack of robust feature selection that transfers well to the test data in the case of adaptive methods. We encourage the reader to refer the accompanying jupyter notebook on this toy dataset for details on the dataset and training.



(a) MNIST, Severity: 1

(b) SVHN, Severity: 1

(c) CIFAR10, Severity: 1

Figure 3-2: Perturbed training image samples obtained by the CMA-ES algorithm for single-pixel perturbation. The perturbed pixel locations are highlighted with red rectangles in the top five rows of the images.

3.6 Benchmarking Optimizers on High Dimensional Training Perturbations

While the above toy dataset analytically demonstrates that adaptive gradient-based methods are not robust to such well-crafted training perturbations, we wish to analyze if this hypothesis also holds for real-world image datasets. As such, we choose popular image datasets of MNIST, CIFAR10, and SVHN for our experiments and generate optimal perturbations for such practical datasets. To benchmark optimizers on high dimensional datasets, we add few-pixel perturbations on training images. Such perturbations are added in a class-specific manner such that all images in the same class are superimposed by the same pixel disturbance pattern. Our goal is to analyze if certain optimizers can ignore such spurious features (with high bias) and learn semantic features like shape and color of the underlying class object that generalize well to test samples. To further benchmark the optimization performance in the presence of worst-case noise, we use the gradient-free optimization technique used in [18] for finding the optimal location and intensity of pixel distribution that maximizes the generalization gap between the train and test samples.

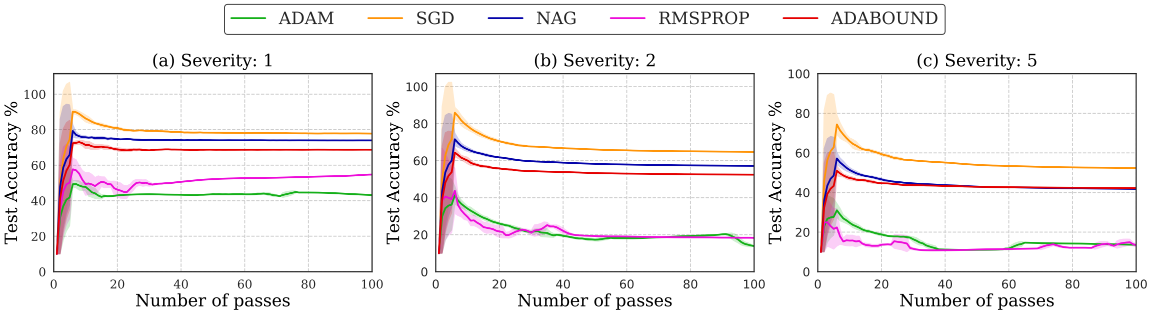


Figure 3-3: Test accuracy on the MNIST dataset when the model is trained on perturbed images with various severity values. SGD and NAG show high test accuracy compared to Adam and RMSProp showing better robustness to training perturbations.

3.6.1 Gradient-Free Perturbation Optimization

Let us assume a classification task with N_c classes. We are interested in finding pixel perturbations that are added in a class-wise fashion, i.e., all images belonging to the same class are perturbed by the same pixel disturbance at N_p spatial locations. Such pixel disturbance is represented as $\Delta = \{\delta_j\}_{j=1,\ldots,N_c}$ with j denoting each class. For class j, the pixel disturbance δ_j is parametrized by the spatial locations and noise intensity of N_p pixels. The goal is to optimize the distribution of pixels Δ, such that the pixel noise is most effective for overfitting the model.

We follow the proposed perturbations objective from [18] that tries to encourage low cross-entropy loss on the perturbed training images and high loss on the clean images to increase the generalization gap, which is represented as a single equation as

$$\max_{\delta,s=0} \min_{\theta,s=1} \mathbb{E}_{(x,y)\ \mathcal{D}} \big[\mathcal{L}_{CE}(x + s\Delta, y; \theta)\big]. \tag{3.12}$$

In the above equation, s acts a switch to turn on/off the perturbation in the training data. The minimization concerning the neural network parameters θ, is performed in the presence of the perturbation, such that it overfits the noise. The maximization is performed with $s = 0$ such that cross-entropy loss on clean samples should be high, thus creating a large generalization gap. According to Equation 3.12, it is difficult to optimize the noise parameter Δ using standard gradient-based methods, because the gradient with respect to Δ is 0 due to multiplication with s. Therefore, CMA-ES [40] is used for the noise optimization.

The fitness score for CMA-ES should encourage high cross-entropy loss on the images from true image distribution while promoting a small loss on the perturbed training samples. This is modeled as the difference of loss terms between these two scenarios which is designated as the semantic mismatch cost (S_m),

$$S_m = \frac{1}{N} \sum_{(x,y)\sim\mathcal{D}} \Big[\mathcal{L}_{CE}(x, y; \theta) - \mathcal{L}_{CE}(x + \Delta, y; \theta)\Big]. \tag{3.13}$$

The above score is maximized by CMA-ES for a fixed θ trained on $(x + \Delta, y)$ samples. The first term encourages a high loss on samples drawn from the true distribution, while the second term promotes a low loss on the perturbed image. This score measures the generalization gap between the samples drawn from true distribution and perturbed distribution, although they differ by only a few pixel changes. The authors of [18] also suggest the use of a domain mismatch score for stable convergence of CMA-ES. Figure 3-2 shows some learned noisy training samples produced by superimposing the learned pixel distributions, which we use to benchmark the robustness of various optimization methods.

3.6.2 CMA-ES based Pixel Noise Optimization

We discuss the details of the CMA-ES algorithm used for finding the worst-case pixel noise, which is added during model training. The objective of the CMA-ES algorithm is to find the parameters (spatial location and pixel intensity) of the pixel disturbance, such that models trained on such noisy samples (a small subset of the original dataset) produce high training accuracy and low testing accuracy on clean samples as outlined in [18]. Only samples from the train-set were used for noise optimization and no test-set samples were used.

Algorithm 3 provides the step by step details of the pixel noise optimization method. The first generation of CMA-ES is started from the initial perturbation parameter $\boldsymbol{\theta}_0 = (\boldsymbol{\mu}_0, \boldsymbol{C}_0)$, where $(\boldsymbol{\mu}_0, \boldsymbol{C}_0)$ are the mean and covariance matrix for the Gaussian distribution used for noise sampling. Multiple pixel perturbation param-

Algorithm 2 CMA-ES based pixel noise optimization

```
Require: Training data $(x, y) \sim \mathcal{D}$, ES params $\boldsymbol{m}_0, \boldsymbol{\Sigma}_0, \sigma_0$
1: for $t$ from 0 to $N_{gen}$ do
2:     Sample noise population: $\{\Delta_j\}_{j=1}^{\lambda} \sim N(\boldsymbol{\mu_t}, \boldsymbol{\Sigma}_t)$
3:     Fit $j^{th}$ models: $\min_\theta \mathbb{E}_{(x,y)\sim\mathcal{D}}[\mathcal{L}_{CE}(x + \Delta_j, y; \theta)]$
4:     Get sematic mismatch $S_m^j = \frac{1}{N} \sum_{(x,y)\sim\mathcal{D}} \Bigg[\mathcal{L}_{CE}(x, y; \theta_j) - \mathcal{L}_{CE}(x + \Delta_j, y; \theta_j)\Bigg]$.
5:     Update ES parameters based on the fitness score, $m_{t+1}, \boldsymbol{\Sigma}_{t+1}, \sigma_{t+1} =$ CMA-ES$(m_t, \boldsymbol{\Sigma}_t, \sigma, \{\mathcal{S}_m^j\})$.
6:     Store the solution with best fitness score in $\delta^*$
7: end for
8: Return best solution: $\delta^*$ as output
```

eters $\{\Delta_j\}_j$ are sampled, for each generation t in step 2. In step 3, we obtain the optimal neural network weights $\boldsymbol{\theta}^*$, by training a CNN from scratch on each such perturbed sample by minimizing the cross-entropy loss using neural network optimization. We used Adam optimizer for this internal DNN training step. However, in Section 3.7.5, we show the results where SGD is used as the internal optimizer. In step 4, we compute the semantic mismatch score by computing the difference in cross-entropy loss between noisy and clean samples. In step 5, after each generation, the sampling parameters are updated by the CMA-ES algorithm to retain the pixel disturbance corresponding to the top-performing costs. We refer the readers to the original paper [40] for more details on the CMA-ES algorithm. The best performing cost across all generations is returned as the output in step 8, which produces the perturbed training images. Severity and number of pixel perturbations are used interchangeably in this chapter.

3.7 Experiments on Image Dataset

After generating the noisy pixel disturbance using CMA-ES, we learn CNN parameters using various optimizers on such noisy train set and evaluate the model on clean test set. For the MNIST dataset, we use a 4-layered CNN model with 2 convolution layers and 2 fully connected layers, while for CIFAR10 and SVHN we used an 8-layered CNN for learning the model. No explicit regularization is used to train

Table 3.2: Network architecture for custom model used for MNIST datasets.

Custom Model for MNIST			
Layer name	**Stride**	**Activation**	**Layer output size**
Input	-	-	$1 \times 1 \times 28 \times 28$
Conv 3×3	1	ReLU	$1 \times 32 \times 26 \times 26$
Conv 3×3	1	ReLU	$1 \times 64 \times 24 \times 24$
Max-Pooling	2	-	$1 \times 64 \times 12 \times 12$
MLP-128	1	ReLU	1×128
MLP-10	1	SoftMax	1×10

Table 3.3: Network architecture for custom model used for CIFAR10/SVHN datasets.

Custom Model for CIFAR10/SVHN			
Layer name	**Stride**	**Activation**	**Layer output size**
Input	-	-	$1 \times 3 \times 32 \times 32$
Conv 3×3	1	ReLU	$1 \times 32 \times 32 \times 32$
Conv 3×3	1	ReLU	$1 \times 32 \times 32 \times 32$
Max-Pooling	2	-	$1 \times 32 \times 16 \times 16$
Conv 3×3	1	ReLU	$1 \times 64 \times 16 \times 16$
Conv 3×3	1	ReLU	$1 \times 64 \times 16 \times 16$
Max-Pooling	2	-	$1 \times 64 \times 8 \times 8$
Conv 3×3	1	ReLU	$1 \times 128 \times 8 \times 8$
Conv 3×3	1	ReLU	$1 \times 128 \times 8 \times 8$
Max-Pooling	2	-	$1 \times 128 \times 4 \times 4$
MLP-512	1	ReLU	1×512
MLP-10	1	SoftMax	1×10

the MNIST dataset, whereas data augmentation consisting of random rotation and cropping was used for training CIFAR10 and SVHN models. Learning rates of 0.01 was used for SGD and NAG. For Adam and RMSProp, learning rate was set at 0.001. These are default values in standard deep learning libraries. During CMA-ES optimization for pixel disturbance, we used Adam optimizer for the internal neural network optimization.

Added to the comparison of robustness of SGD-based methods compared to adaptive optimizers, we also discuss more results on the image datasets to demonstrate how our observations vary across different hyper-parameter settings. Specifically, we ask the following questions: (1) Does changing the learning rate improve the robustness of the studied optimizers to training noise? (2) Does adaptive gradient methods perform better than SGD-based methods if the CMA-ES algorithm is tuned to find pixel disturbance for SGD optimizer in the internal CNN optimization step? (3) Is a similar performance observed on commonly occurring noise, which has not been tuned by a gradient-free optimization method? Additionally, we provide the training accuracy curves to show that in most cases, the models achieve 100% training accuracy but low testing accuracy exhibiting signs of overfitting to the training noise.

3.7.1 Experimental setup

We performed all the experiments on Ubuntu 16.04 on Nvidia Quadro RTX 8000 GPUs. For the experiments shown, we report the mean of three independent runs with a moving average on the accuracy curve of window size 5. We report the results on custom CNN models only for MNIST and CIFAR10/SVHN. We additionally show results for ResNet models on MNIST dataset as well.

Custom model for MNIST: This is a simple CNN structure with 2 Conv layers and 2 MLP layers which is designed to take gray-scale image inputs of shape $(1 \times 28 \times 28)$. The architecture is given in Table 3.2.

Custom model for CIFAR10/SVHN: This CNN has 6 Conv layers and 2 fully connected layers which are designed to take color image inputs of shape $(3 \times 32 \times 32)$. The architectural details are provided in Table 3.3.

ResNet: In addition to the above models, we also test with ResNet18 and ResNet50 model [42]. We changed the first convolution layer to feed in the grayscale image for the MNIST dataset. These models have significantly more parameters and are much more liable to overfitting. However, our experiments show that SGD and NAG exhibit good robustness to the training noise which prevents overfitting even in a highly over-parametrized setting.

3.7.2 Quantitative Analysis of Test Accuracy

We present the test accuracy of various optimizers trained in the presence of pixel noise. The training curves for most cases correspond to 100% accuracy at final epoch.

MNIST: Figure 3-3 shows the performance of various optimization with three severity levels. The curves are averaged from three independent runs starting from random model parameters. For all optimizers, test accuracy initially rises to a relatively high value after which it gradually decreases signifying overfitting to the pixel perturbations. While adaptive gradient methods such as Adam and RMSProp show low final test accuracy of 43% and 55% respectively, even for single-pixel noise, SGD and NAG show comparatively higher resilience to overfitting with 78% and 74% accuracy respectively. Adabound [62] uses strategies from both SGD and Adam, thus showing intermediate performance. Overfitting increases with increasing severity of the pixel noise, however, SGD and NAG show better robustness for all cases. Stronger performance of SGD optimizers without overfitting is also shown based on learning rate tuning.

CIFAR10: Figure 3-4 shows a comparison of test accuracies of four optimizers trained on the CIFAR10 dataset in the presence of pixel training perturbations. Data augmentation was used in the training phase. We observe a similar pattern where SGD and NAG show better robustness compared to adaptive gradient methods. For severity 1, NAG shows performance at par with SGD, whereas for severity 2, NAG reaches a peak test accuracy after which it starts overfitting. Vanilla SGD shows very strong performance without signs of overfitting to the pixel perturbations as shown by the monotonically increasing nature of the test accuracy curve. In the presence of early stopping regularization [17], NAG has the potential to show better test accuracy at an early stage compared to SGD.

SVHN: Similar training settings as CIFAR10 was used in this case as well. Figure 3-5 shows the performance of various optimization on the SVHN dataset in the presence of training noise. Vanilla SGD shows a very strong performance in this case with final test accuracy reaching almost 93% and 92% which is close to clean training

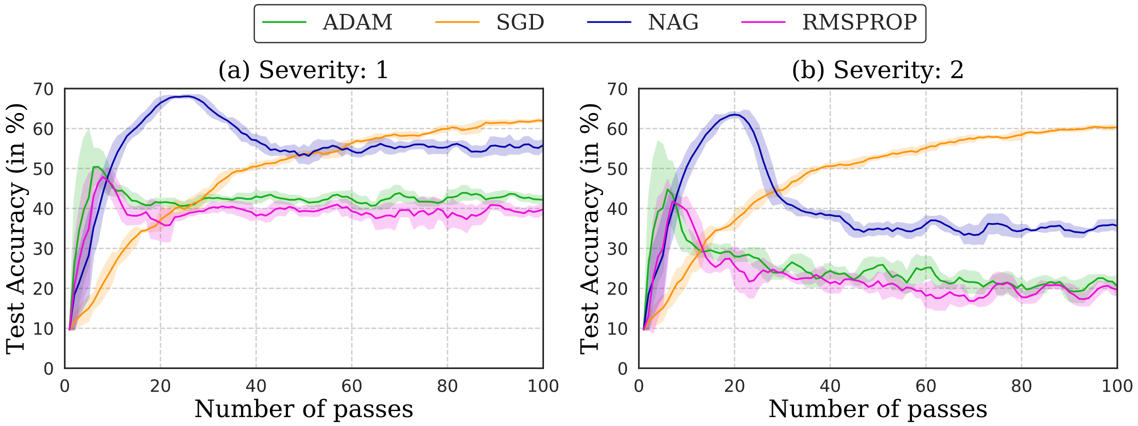


Figure 3-4: Test accuracy on CIFAR10 dataset on CNN trained on perturbed images. Vanilla-SGD methods show high test accuracy compared to adaptive methods.

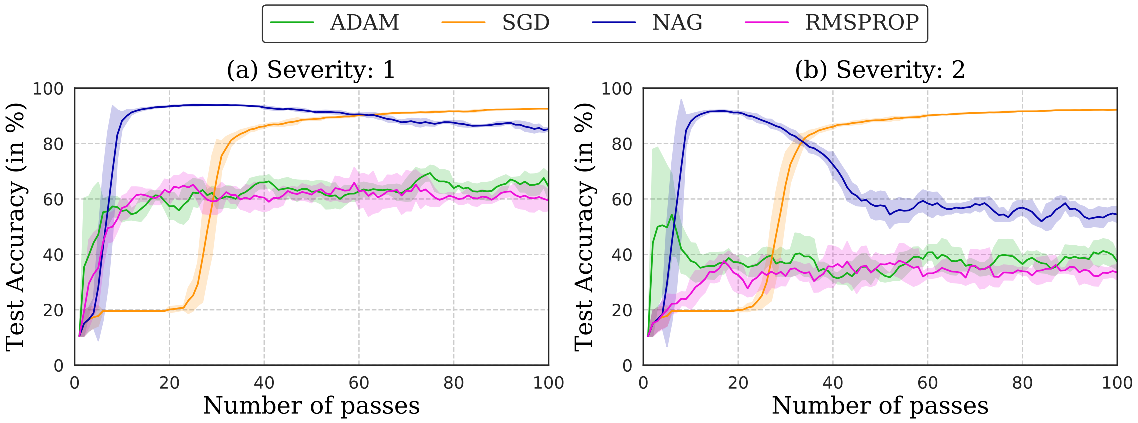
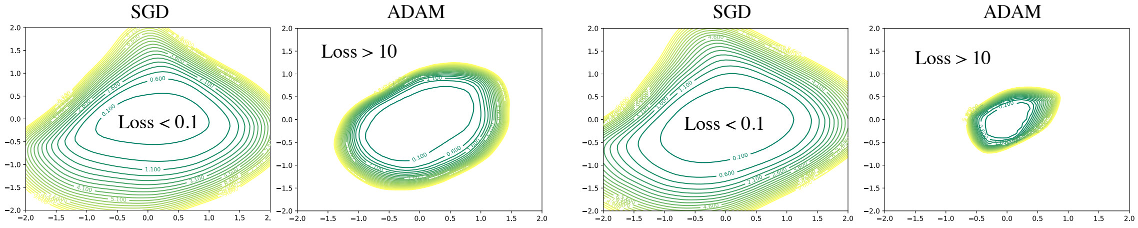


Figure 3-5: Test accuracy when trained on perturbed SVHN datasets.

accuracy. Other optimizers show signs of overfitting after slightly higher initial test accuracy similar to the CIFAR10 case.

3.7.3 Qualitative analysis of 2D Loss Surface

To understand the behavior of the local minima reached by optimization methods, we plot the loss surface around the solution obtained by SGD and Adam, as shown in Figure 3-6, which shows iso-contours of loss surfaces ranging from values from 0.1 to 10.0, based on the work of [60]. For all cases, we use the same scale of parameters within $(-2, 2)$ units from the final solution. Qualitative analysis of the loss surface shows that SGD finds a local minimum which is comparatively flatter compared to



(a) MNIST, Severity: 1

(b) MNIST, Severity: 2

Figure 3-6: Comparing the 2D loss contour for SGD and Adam around the learned model parameters at the same scale. Loss iso-contours are from 0.1 to 10.0. The white region at the center shows loss values less than 0.1 and the region outside the yellow contour shows loss values more than 10. Adam tends to find solution around valleys with steeper gradients whereas SGD finds a solution which is comparatively flatter.

the minima found by Adam. The loss contours for Adam rapidly change its value from 0.1 to 10, showing a steep descent into the valley. Previous works [45, 51] argue that solutions that lie in a sharper minimum tend to generalize worse which might contribute to the overfitting nature of adaptive methods.

The presence of optimal pixel disturbances might induce such sharp minima on the loss surface which adaptive methods typically converge upon. We find that the sharpness of the minima obtained by adaptive gradient methods increases with noise severity, whereas the minima reached by SGD shows negligible changes in iso-contour distribution between the two severity levels. We believe adaptive methods get stuck in such deep wells from which it cannot escape due to diminishing value to the update step size caused by the inverse of accumulated gradient squares in Equation 3.6. On the other hand, even if SGD falls in such deep wells, due to non-adaptive step size there is a good chance it can escape converging to a final flatter region.

3.7.4 Effect of Learning rate on Noise Robustness

We performed model training for various learning rates (lr) from the set, $0.1, 0.01, 0.001$, and 0.0001. In the main text, $lr = 0.01$ was used for SGD and NAG, and $lr = 0.001$ was used for Adam and RMSProp, which are default values in standard deep learning libraries. The results for test accuracy for custom CNN are shown in Figure 3-7, Fig-

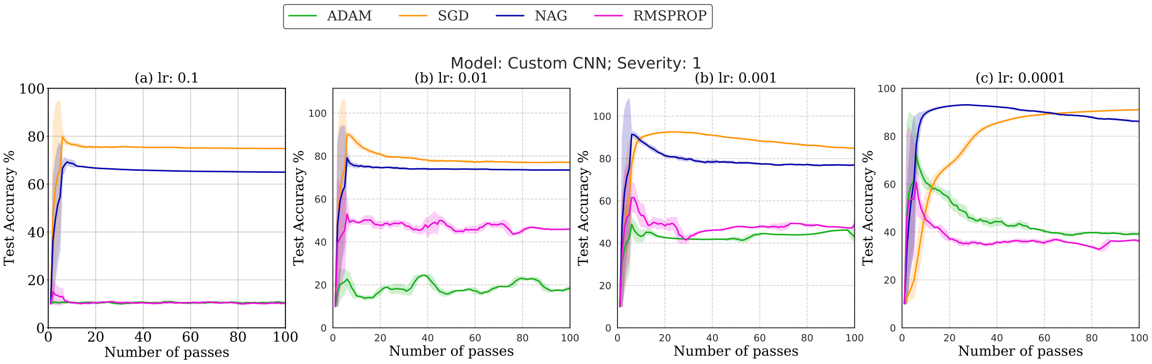


Figure 3-7: Test accuracy on MNIST for custom model with noise severity (number of pixel noise)=1 for various learning rates. SGD and NAG show better performance for lower learning rate of 0.0001 compared to default values of learning rate.

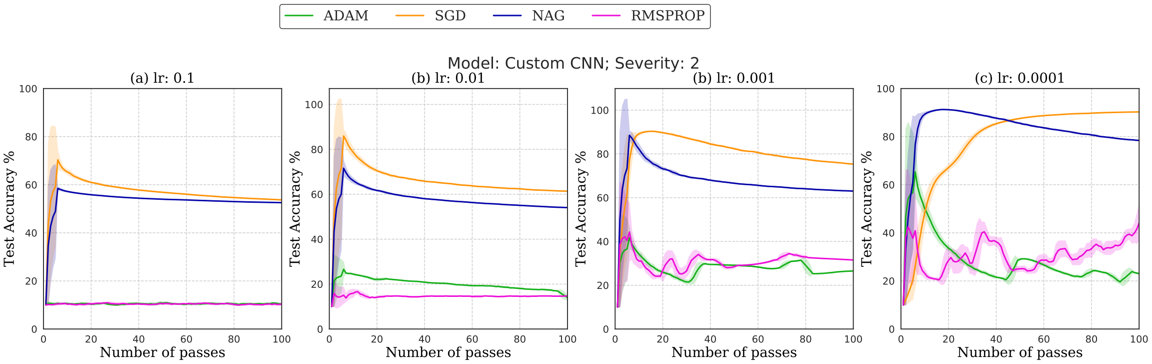


Figure 3-8: Test accuracy on MNIST for custom model with noise severity (number of pixel noise)=2 for various learning rates. SGD and NAG show better performance for lower learning rate of 0.0001 compared to default values of learning rate.

ure 3-8, and Figure 3-9. For learning rates of 0.1 and 0.01, Adam and RMSProp could not converge during training. For other learning rates, the training accuracy was close to 100% for most cases. The test accuracies show that the learning rate of 0.0001 exhibits significantly more robustness to the training noise compared to the default learning rate values. For example, SGD trained with $lr = 0.0001$, shows final test accuracy of 91%, 90%, and 83%, compared to 76%, 61%, and 48% for SGD trained with $lr = 0.01$, for severity 1, 2, and 5 respectively. While all methods demonstrate better testing accuracies, SGD and NAG particularly show good testing accuracies compared to adaptive gradient methods.

The test accuracies for ResNet18 are shown in Figure 3-10, Figure 3-11, and Figure 3-12 and test accuracies for ResNet50 are shown in Figure 3-13, Figure 3-14,

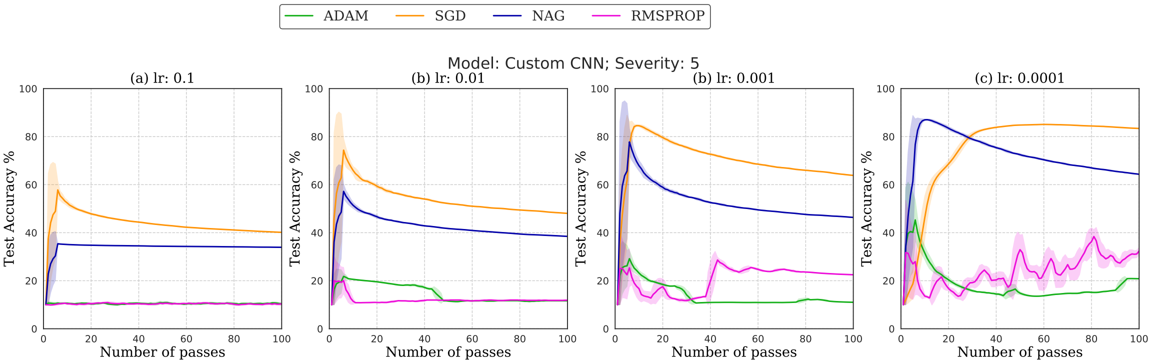


Figure 3-9: Test accuracy on MNIST for custom model with noise severity (number of pixel noise)=5 for various learning rates. SGD and NAG show better performance for lower learning rate of 0.0001 compared to default values of learning rate.

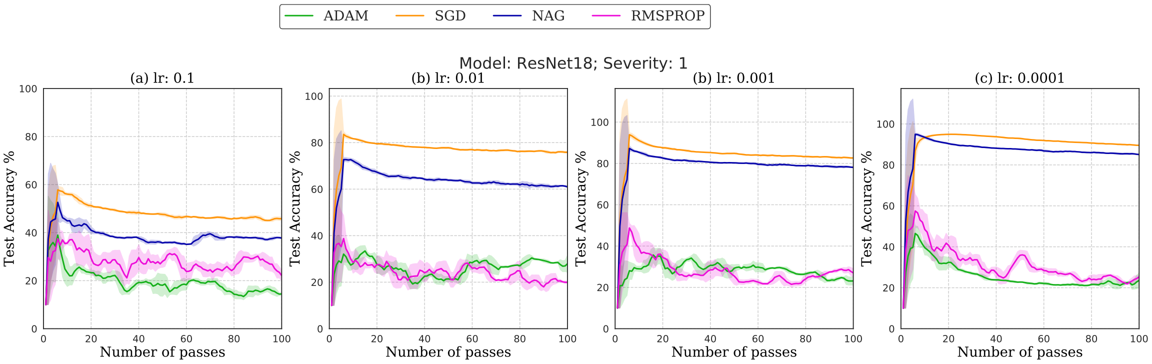


Figure 3-10: Test accuracy on MNIST for ResNet18 model with noise severity (number of pixel noise)=1 for various learning rates.

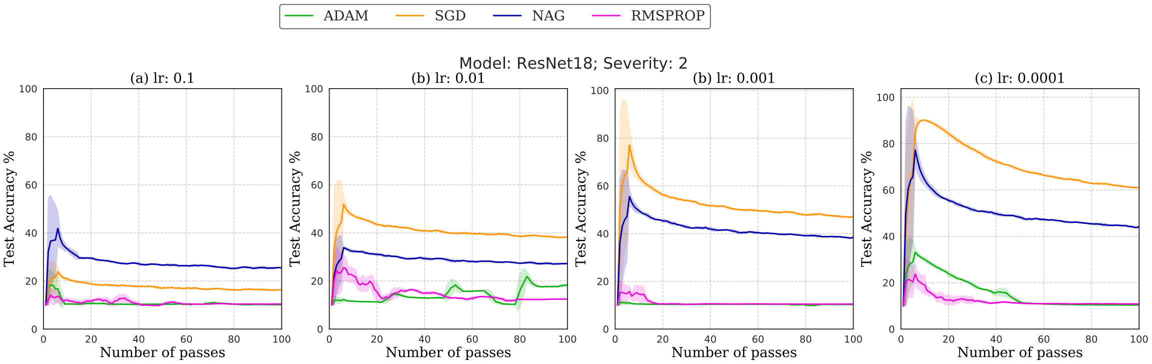


Figure 3-11: Test accuracy on MNIST for ResNet18 model with noise severity (number of pixel noise)=2 for various learning rates.

and Figure 3-15. Since these models have more trainable parameters, they are more prone to overfitting. However, we observe similar results that SGD-based methods are more robust compared to adaptive methods. Furthermore, $lr = 0.0001$ shows

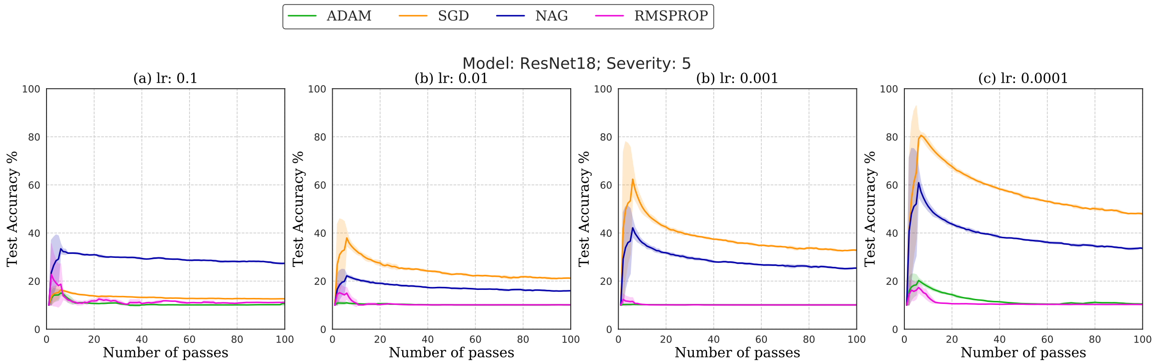


Figure 3-12: Test accuracy on MNIST for ResNet18 model with noise severity (number of pixel noise)=5 for various learning rates.

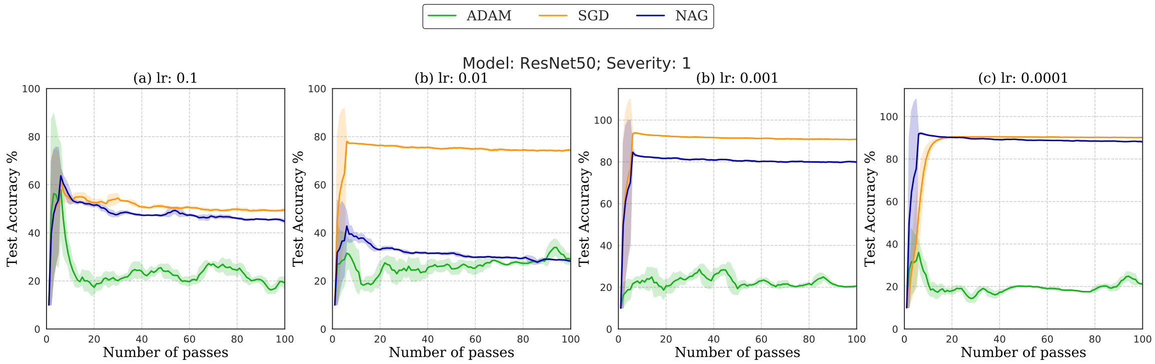


Figure 3-13: Test accuracy on MNIST for ResNet50 model with noise severity (number of pixel noise)=1 for various learning rates.

best performance amongst all the settings.

3.7.5 Pixel noise with SGD-based CMA-ES optimization:

During pixel noise optimization in Algorithm 3, step 3 uses a CNN training stage. We used Adam optimizer in this step, which might produce noise specifically for Adam optimizer. Additionally, we also performed pixel noise optimization where SGD was used in the internal CNN training step for CMA-ES. We trained the above-discussed models with such pixel noise disturbance specifically trained against the SGD optimizer. The corresponding testing accuracies are shown in Figure 3-16, Figure 3-17, and Figure 3-18. Even in this setting, we observe that SGD-based methods outperform the adaptive gradient methods. Therefore, SGD-based methods exhibit better noise robustness which is agnostic of the internal optimizer used for the CMA-ES

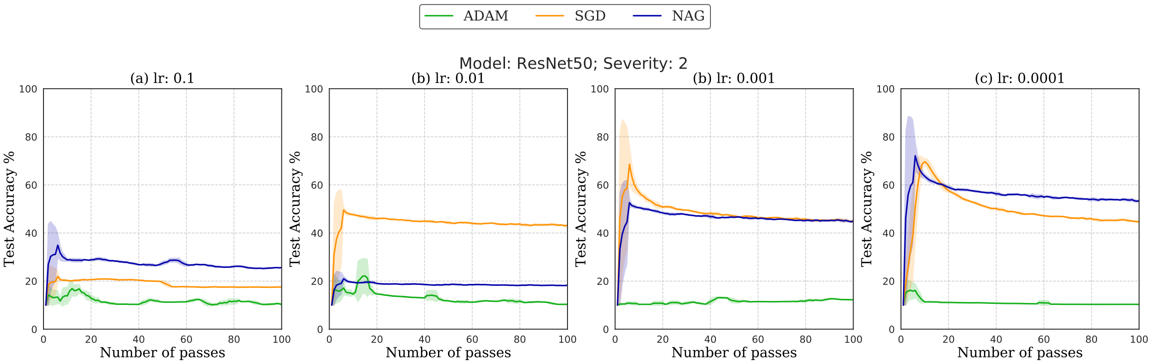


Figure 3-14: Test accuracy on MNIST for ResNet50 model with noise severity (number of pixel noise)=2 for various learning rates.

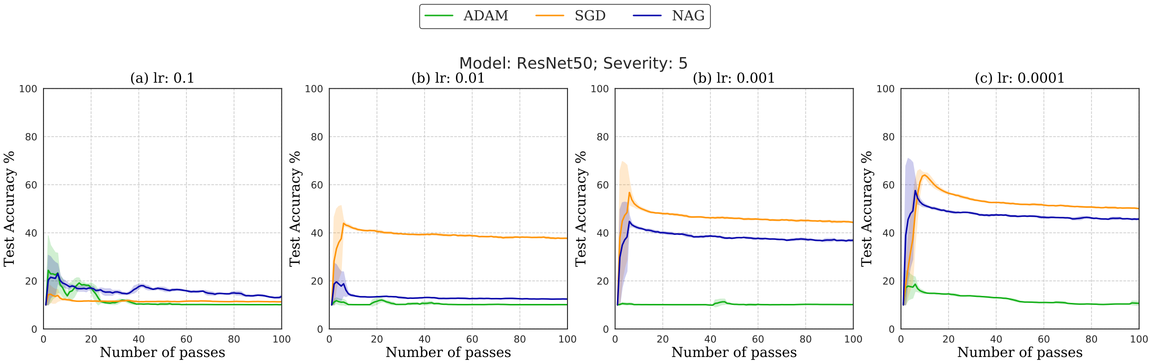


Figure 3-15: Test accuracy on MNIST for ResNet50 model with noise severity (number of pixel noise)=5 for various learning rates.

noise generator.

3.7.6 Training with Impulse Noise:

While most of the results demonstrate the test accuracy of various optimizers for worst-case noise that was obtained by gradient-free parameter optimization, in this experiment, we also show that our observation holds for commonly occurring noise patterns that are not obtained after special optimization procedure. As such, we use the impulse noise defined in the work of [43] on the MNIST dataset and train models with various optimizers. The testing accuracies for impulse noise are provided in Figure 3-19, which also demonstrates that SGD and NAG exhibit better robustness compared to adaptive gradient methods.

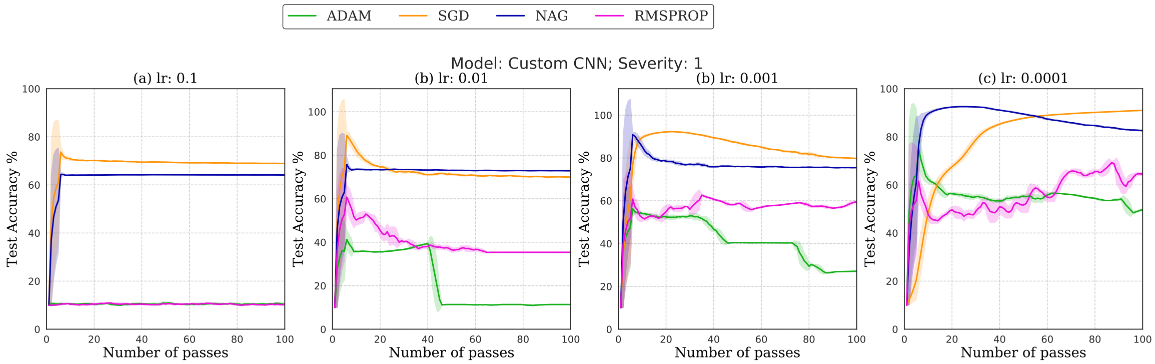


Figure 3-16: Test accuracy on MNIST for custom CNN model with pixel noise trained by CMA-ES with SGD as internal optimizer (severity=1).

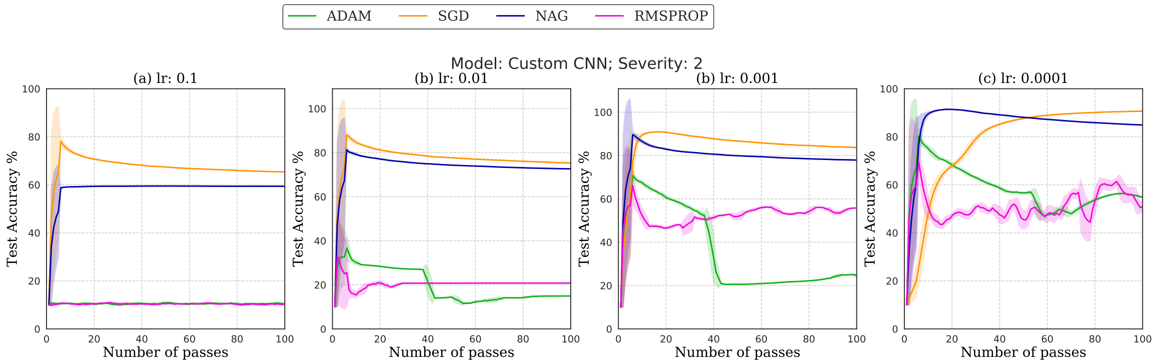


Figure 3-17: Test accuracy on MNIST for custom CNN model with pixel noise trained by CMA-ES with SGD as internal optimizer (severity=2).

3.8 Discussion

Relevance to real-world noise: Although we benchmark the optimization methods in the presence of worst-case perturbations, the results presented here also apply to general noise distribution that is not optimized by CMA-ES. We show results on optimization robustness against uniformly sampled "impulse" noise with pixel disturbances at arbitrary spatial locations. The results highlight similar robustness of SGD over adaptive methods in the presence of training noise. Additionally, [105] has shown that even it the absence of training noise, SGD with proper learning rate tuning can outperform adaptive gradient methods.

SGD-based perturbations: One might argue that the crafted noise signals for both toy-problems and high-dimensional data problems were designed to fool adaptive gradient methods, it might be possible to find noise signals that cause overfitting in

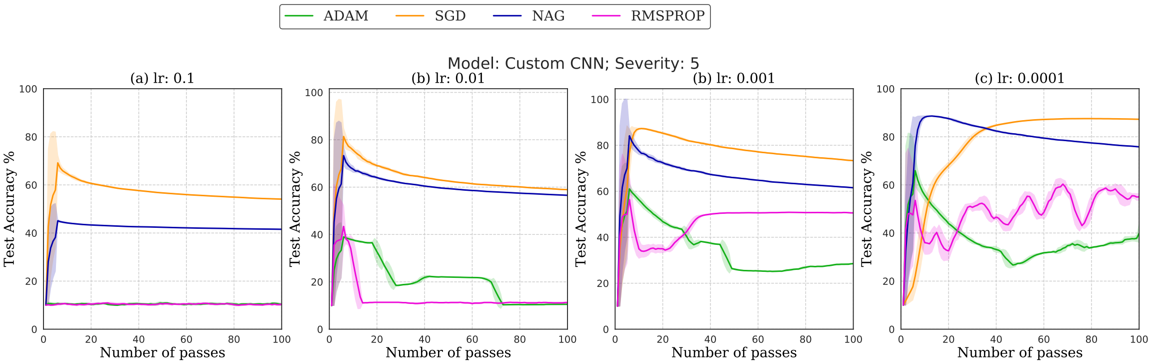


Figure 3-18: Test accuracy on MNIST for custom CNN model with pixel noise trained by CMA-ES with SGD as internal optimizer (severity=5).

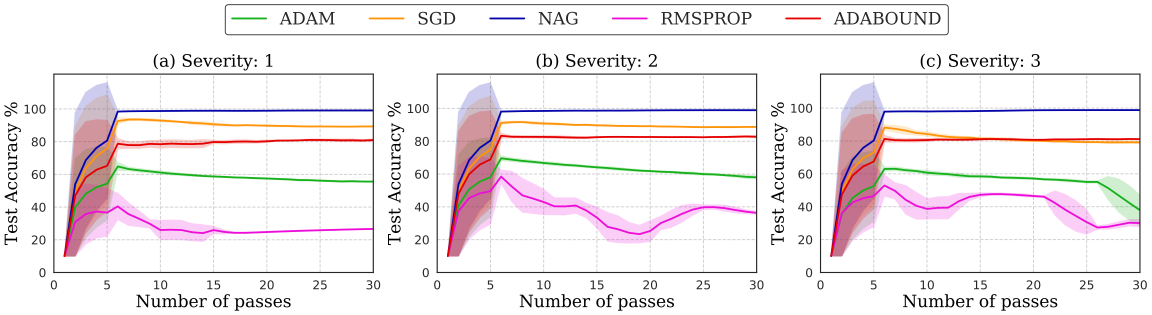


Figure 3-19: Test accuracy on MNIST for custom model with impulse noise following the implementation in [43] for different severity levels.

SGD and zero test error on adaptive methods. While it might be possible to construct such datasets, that cause unevenness in the loss terrain where SGD based methods might have difficulty in navigating to the local minimum, in such a case, both training and the testing accuracy will be poor which results in underfitting and not overfitting which is the focus of this study. We also find that careful learning rate tuning can make learning possible in such cases as well. However, the converse is not true for adaptive methods trained under noisy data because such methods tend to have an adaptive and diminishing update step sizes. Therefore, irrespective of the starting learning rate, they are liable to get stuck in steep loss structures. Additionally, we show the results of benchmarking optimizers under noise perturbations trained by CMA-ES with SGD as the internal optimizer which also illustrates the superior robustness of SGD over adaptive methods.

3.9 Conclusions

In this work, we present several pieces of analytical and experimental evidence that adaptive gradient methods are not well-suited for optimizing neural networks in the presence of training noise. Instead, vanilla stochastic gradient methods show better robustness to such perturbations during training. We present an artificial toy dataset, on which we illustrate that adaptive methods show poor generalization in the presence of training noise while SGD optimizer shows perfect classification. We also confirm our hypothesis on high dimensional image datasets where SGD and its variants show better noise robustness compared to adaptive methods. Since adaptive methods use a history of past gradients, these methods tend to get stuck in a steep local optimum solution that shows over-reliance on these spurious features, from which they cannot recover due to diminishing update rates. In contrast, SGD methods find the minimum ℓ_2 norm solution and avoid falling into such steep local minima due to the fixed step size, thereby inducing an implicit regularization framework for noise robustness. In this work, we focus on classification tasks for the benchmarking exercise. In the future, we aim to study the robustness of such optimizers on generative and unsupervised tasks.

Chapter 4

Deep Learning for Medical Images under Adversarial Training Attacks

Adversarial examples in deep learning systems have serious practical implications on the usability of such methods for safety-critical applications like medical image analysis. While most previous works in this domain study vulnerabilities for testing time perturbations, for the first time, we study the effect of training distribution shifts on out-of-sample generalization for medical image datasets. We utilize Evolution Strategy (ES) based benign few pixel adversarial perturbation generation algorithm, to corrupt the training samples and measure generalization performance on clean testing samples. Empirical evaluations demonstrate that significantly low test accuracy (with almost 100% train accuracy) can be achieved on two medical image classification datasets, by just perturbing a single pixel in the training images. We benchmark input covariate shift normalization and the effect of various optimization techniques which reveal that vanilla Stochastic Gradient Descent (SGD) methods are more robust than popular adaptive gradient techniques (like ADAM) under such pixel-based training distribution shifts. Therefore, our method cautions practitioners to benchmark their models using various optimization methods in the presence of training perturbations.

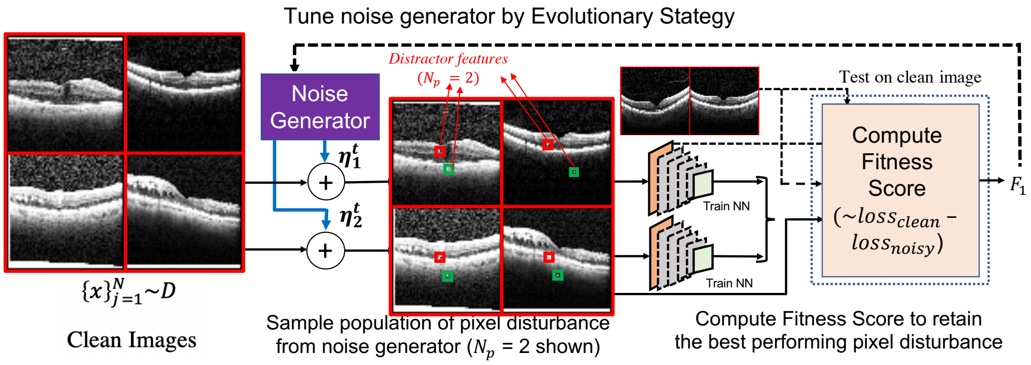

Tune noise generator by Evolutionary Stategy
Noise Generator
η_1^t
η_2^t
Distractor features (N_p = 2)
Test on clean image
Compute Fitness Score ($\sim loss_{clean} - loss_{noisy}$)
F_1
Train NN
Train NN
$\{x\}_{j=1}^{N} \sim D$
Clean Images
Sample population of pixel disturbance from noise generator (N_p = 2 shown)
Compute Fitness Score to retain the best performing pixel disturbance

Figure 4-1: Overview of the evolutionary strategy based pixel perturbation finding algorithm. Noise generator parameters are updated by evolutionary strategy [40] following best performing perturbations that maximizes the fitness score.

4.1 Introduction

The notable empirical success of deep learning models in various application areas like computer vision [42, 85], natural language processing [23, 79, 104] and other real-world domains, is poised to lead us to the next industrial revolution. Yet despite their overwhelming commercial success, current deep learning systems are not robust [6, 55, 91, 98] against adversarial examples. Such adversarial vulnerabilities are especially detrimental in safety-critical applications like medical image analysis thus requiring extensive robustness studies under such weaknesses. While most existing methods on adversarial attack and defense, study distribution shift (as imperceptible perturbations) during test time, there is a limited study of vulnerabilities due to distribution shifts on training samples.

Typically, adversarial attack algorithms [16,35,63,98] use a trained neural network to generate small imperceptible perturbations on *adversarial* query images that result in false classification. In the medical image analysis domain, [29] perform analysis of various adversarial techniques like Projected Gradient Descent (PGD), adversarial patches, etc. on Fundoscopy, Chest X-ray and Dermoscopy datasets. Ozbulak et al. [74] performed the analysis of biomedical segmentation under adversarial settings, while Paschali et al. [77] showed the performance of classification and segmentation under adversarial data, noisy settings, and ambiguous input data.

While evasion adversarial attacks only study distribution shifts on testing data, data poisoning attacks [12, 91, 96] are a variety of adversarial attacks, where the adversary injects a few malicious samples in the training data to cause incorrect classification (typically targeted) on few clean test data samples. In the neural network regime, popular poisoning methods use back-gradient optimizations [67, 91] or influence functions [55].

In this work, we expose the vulnerability of deep learning models for analyzing medical images under *worst-case* few pixel perturbations on training images. Let us consider a practical case scenario where a dot/few dots (almost imperceptible to human eyes) have appeared on the medical image because of some device noise. In this study, we show that if the model is trained using those single/few pixels perturbed images, the network learns absolutely nuisance features instead of useful semantic features and provides unexpectedly low test accuracy. Thus, we claim that studying model robustness under such training sample perturbations is of practical importance from the safety-critical point-of-view. Such training time noise can also be intentionally put by a malicious agent to sabotage the model. To train our model on the perturbed dataset, we perform adversarial distribution shifts by adding a few pixels to the training images. However, to keep it imperceptible, we restrict these pixel distribution shifts to just a few pixel changes (N_p= 1 to 5) on the training samples. Testing samples are kept clean. It is to be noted that, this is different from poisoning attacks where only a few samples are modified for back-door attacks without affecting the performance of most test time images.

4.2 Methodology

In this section, we describe the method used for adversarial pixel perturbation generation utilizing evolution-based optimization on medical images, and elaborate the input Covariate Shift Normalization approach for improving out-of-sample generalization.

4.2.1 Evolution-based Training Perturbation Optimization

We consider an input space of images $X \in \mathbb{R}^N$ and corresponding label space $Y \in \{0,1\}^C$, where C is the number of classes. The true data distribution is given as $(\boldsymbol{x}, y) \sim \mathcal{D}$ and the adversarial shifted training distribution, $(x_{adv}, y) \sim \mathcal{D}_{adv}$ is obtained from the images $\boldsymbol{x}$ by adding few pixel noise. We choose the standard classification task with neural network, $F_\theta(x)$ producing the probability of output classes. Our objective is to find the optimal pixel corruptions to train the classification model, such that the classification performance on clean samples are low.

Following [19], we design pixel perturbations to encode class-specific information. As such, given a noise sample, δ, all images in class k will have the pixel noise pattern (δ_k). For ease of representation, we show the corruption of training images by the addition operation $(x + s\delta)$. We solve a min-max problem of the form

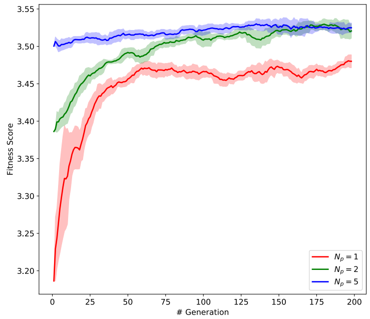
$$\max_{\substack{\boldsymbol{\delta} \\ s=0}} \min_{\substack{\boldsymbol{\theta} \\ s=1}} \mathbb{E}_{(x,y)\sim\mathcal{D}}\big[\mathcal{L}_{CE}(x + s\delta, y; \theta)\big], \tag{4.1}$$

where θ is the CNN parameter and s, is the selector variable. The CNN model is trained in the presence of noise (with $s = 1$) by minimizing the cross-entropy loss $\mathcal{L}_{CE}$. Alternatively, we maximize loss on clean samples (with $s = 0$) to find the optimal pixel perturbations. According to Equation 4.1, there is zero gradient with respect to the noise parameter δ. Therefore, it is not straight forward to use gradient-based method for noise optimization. Due to this problem, we resort to a gradient-free evolutionary strategy method for training distribution-shift optimization. Specifically, we use CMA-ES [40], which has been shown to work well in high-dimensional problems [39]. We refer to corruption of training images by such optimal perturbations as "adversarial training distribution shift".

Fitness Score: The CMA-ES algorithm optimizes its tunable parameters to maximize a specific fitness score. The fitness score is designed to encourage high loss on the images from true image distribution when the model is trained on perturbed samples with low loss. Thus, the fitness score measures the difference of cross-entropy loss on clean samples and noisy perturbed samples as follows

	OCT Dataset			
Networks	Clean	N_p=1	N_p=2	N_p=5
Resnet50	0.02	0.60	0.71	**0.75**
DenseNet121	0.01	0.58	0.72	**0.73**
	Derma Dataset			
Resnet50	0.31	0.45	**0.71**	0.46
DenseNet121	0.19	0.56	0.74	**0.80**

(a) Training on images corrupted by evolutionary pixel perturbations have high testing error on clean data even when learned with single pixel noise.



(b) Learning curve for Evolutionary strategy on OCT dataset.

Figure 4-2: (a) Testing error on clean samples when trained on learned pixel perturbed training data (ADAM optimizer). State-of-the-art deep learning models show low testing accuracy on clean samples, (b) Learning curve of the CMA-ES algorithm on OCT dataset show improving fitness score with increasing generations.

$$\mathcal{F}_m^{\delta} = \frac{1}{N} \sum_{(x,y)\sim\mathcal{D}} \Big[\mathcal{L}_{CE}(x, y; \theta) - \mathcal{L}_{CE}(x + \delta, y; \theta) \Big]. \tag{4.2}$$

In addition, we also use a domain divergence based fitness score as mentioned in [19], $\mathcal{F}_d^{\delta}$ which ensures that noisy and clean samples have high linear separability to maximize the chance of over-fitting.

We present details of the optimal pixel disturbance search method in Algorithm 3. For each generation t, we sample pixel noise $\{\delta_j\}_j$ and obtain the optimal neural network weights $\boldsymbol{\theta}^*$, by training a CNN from scratch on each such noisy training samples. After each generation, the sampling parameters are updated by the CMA-ES algorithm to retain the pixel noise corresponding to the top-performing fitness

Algorithm 3 Evolutionary Pixel Distribution Shift

```
Require: Train data $(x, y) \sim \mathcal{D}$, Initial ES parameters $\boldsymbol{m}_0, \boldsymbol{\Sigma}_0, \sigma_0$
for $t$ from 1 to $N_{gen}$ do
    Sample a population of noise: $\{\delta_j\}_{j=1}^{\lambda} \sim N(\boldsymbol{\mu_t}, \boldsymbol{\Sigma}_t)$
    Fit $j^{th}$ models $F_\theta^j$ : $\min_\theta \mathbb{E}_{(x,y)\sim\mathcal{D}}[\mathcal{L}_{CE}(x + \delta_j, y; \theta)]$
    Compute fitness for $j^{th}$ noise sample, $\mathcal{F}_j = \mathcal{F}_m^{\delta_j} + \mathcal{F}_d^{\delta_j}$.
    Update ES parameters $m_{t+1}, \boldsymbol{\Sigma}_{t+1}, \sigma_{t+1} = \text{CMA-ES}(m_t, \boldsymbol{\Sigma}_t, \sigma, \{\mathcal{F}_j\})$.
    Store the solution with best fitness score in $\delta^*$
end for
```

Table 4.1: Precision/Recall/F1 score for each class in OCT and Derma dataset for DenseNet-121 (ADAM). High values are bold and low values are shown in red.

	OCT Dataset			
	Clean	N_p=1	N_p=2	N_p=5
NORMAL	**1.00/0.97/0.99**	0.54/0.7/0.61	0.24/0.44/0.31	0.00/0.00/0.00
DRUSEN	**0.98/1.00/0.99**	0.00/0.00/0.00	0.33/0.50/0.39	0.35/0.21/0.27
CNV	**1.00/1.00/1.00**	0.36/0.96/0.52	0.28/0.09/0.14	0.33/0.01/0.02
DME	**0.99/1.00/1.00**	0.25/0.01/0.02	0.31/0.15/0.20	0.26/0.90/0.40
	Derma Dataset			
	Clean	N_p=1	N_p=2	N_p=5
Pigmented Macule	**0.69/0.66/0.67**	0.28/0.81/0.42	0.33/0.09/0.14	0.21/0.33/0.25
Erythema	**0.88/0.87/0.87**	0.81/0.32/0.46	0.75/0.28/0.40	0.65/0.16/0.25
Ulcer	**0.89/0.82/0.85**	0.71/0.56/0.63	1.00/0.19/0.33	1.00/0.03/0.05
Tumor	**0.56/0.71/0.63**	0.33/0.21/0.26	0.09/0.91/0.17	0.08/0.75/0.14
Leukoderma	**0.62/0.80/0.70**	0.24/0.44/0.31	0.00/0.00/0.00	0.00/0.00/0.00

score [1]. The best performing pixel perturbation by fitness score across all generations is returned as the optimal perturbation.

4.2.2 Input Covariate Shift Normalization

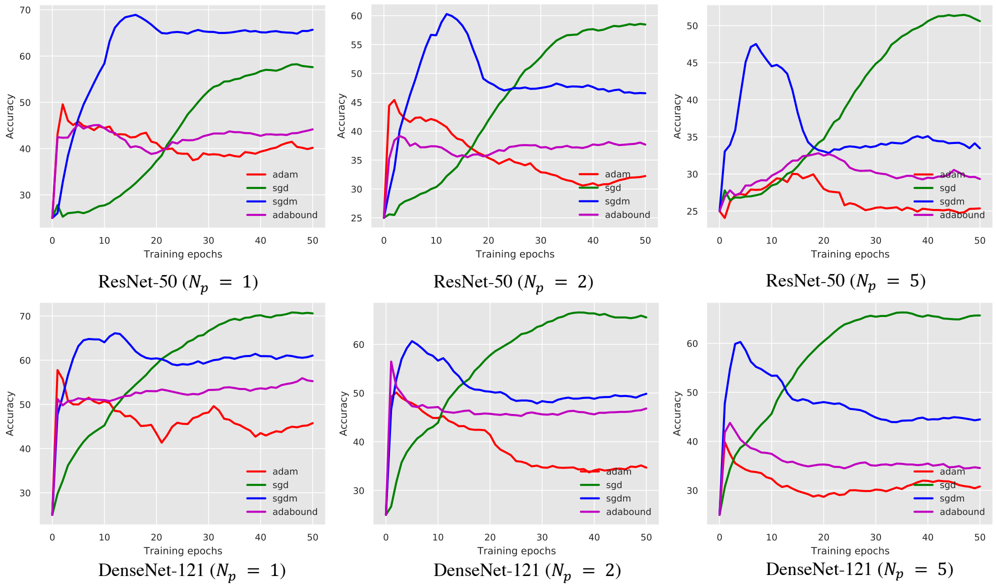
The above method for adversarial distribution shift generation causes a high loss of generalization due to the distribution shift between training and testing samples. We use concepts from BatchNormalization [47] to reduce the covariate shift between training and testing distributions. Therefore, we normalize the image data to zero mean and unit standard deviation for each batch, $x^b_{normed} = \frac{x^b - \mu^b}{\sigma^b}$, both for training and testing. We refer to this as input Covariate Shift Normalization (CSN). We benchmark deep models on medical image datasets with/without CSN as described in the experimental section later.

4.3 Experiments

Medical Image Dataset

To evaluate our method, we use two different medical image classification dataset, (a) Retinal Optical Coherence Tomography (OCT) images [50] (b) Dermatological images

[1] More details on the CMA-ES algorithm can be found in the original paper [40]



ResNet-50 ($N_p = 1$) ResNet-50 ($N_p = 2$) ResNet-50 ($N_p = 5$)

DenseNet-121 ($N_p = 1$) DenseNet-121 ($N_p = 2$) DenseNet-121 ($N_p = 5$)

Figure 4-3: Test accuracy curves for CNN training under pixel disturbance on OCT dataset shows vanilla SGD based methods generalize better compared to adaptive optimization methods as shown by higher testing accuracy.

for east asian race [65]. The first dataset consisted of gray-scale OCT images provided by [50] which covered classification over four categories: Normal, Drusen's Syndrome, Choroidal Neovascularization (CNV), and Diabetic Macular Edema (DME). We resized each image in the dataset to 64 × 64 dimensions. We used a subset of the dataset for training with 8616 images for training and 242 unseen images for testing. The second dataset consists of dermatological images collected from volunteers belonging to the East Asian race [65]. [2]. It consists of color images usually larger than 200 × 200 pixels. We choose five classes from the original dataset: (i) Pigmented Macule, (ii) Erythema, (iii) Ulcer, (iv) Tumor, and (v) Leukoderma. To handle class imbalance, we used oversampling resulting in 3078 training images per class and 1344 total validation images. Each image was resized to 64 × 64 × 3.

Deep Neural Network Architectures

We study the robustness of two deep image classification models: (i) ResNet-50 [42] which uses "identity shortcut connections" to solve the vanishing gradient problem

[2] Since the dataset is not available publicly, we obtained the dataset by contacting the authors

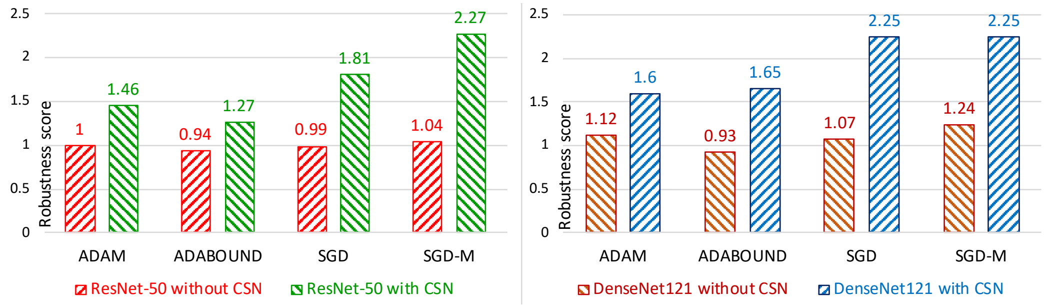


Figure 4-4: Robustness score of various deep models and optimization pairs with/without CSN. SGD-based methods are more robust compared to adaptive methods and generalize better in presence of training noise. Input normalization (CSN) improves model performance especially for SGD-based methods.

in very deep models, and (ii) DenseNet-121 [46] which improves upon the ResNet architecture by connecting all layers with each other in "Dense" group. We evaluate these models with and without CSN for 4 optimization methods: ADAM [53], Adabound [62], SGD, and SGD with momentum (SGD-M).

Evaluation metric

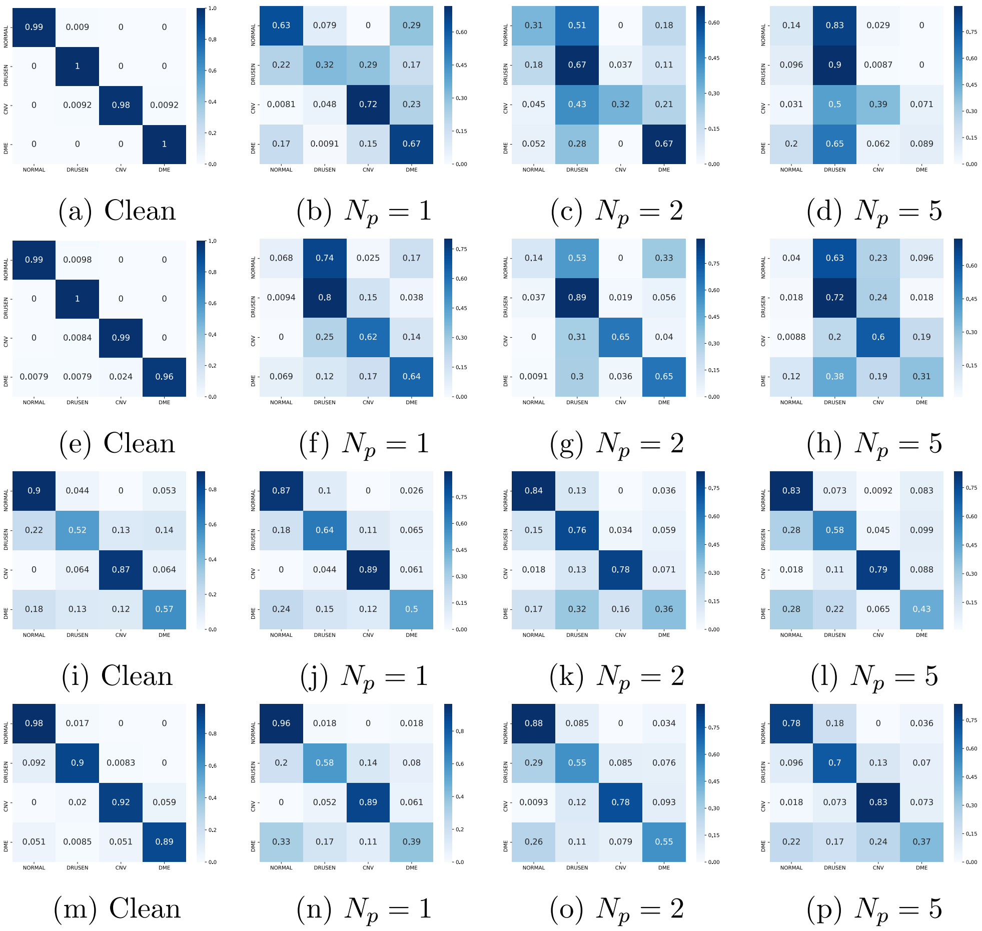
We use standard testing error/accuracy along with precision, recall, and F1-score for each class, to benchmark various deep learning models. To obtain an aggregated measure of robustness, we propose a standardized *robustness score*, $\mathcal{R}_m^{\text{otp}}$ to measure the performance of various models (m) and optimization techniques (opt) with respect to a common base model and optimization setting, given as

$$\mathcal{R}_m^{\text{otp}} = \Bigg(\sum_{N_p=\{1,2,5\}} \text{acc}_m^{opt} \Bigg) \Bigg/ \Bigg(\sum_{N_p=\{1,2,5\}} \text{acc}_{\text{ResNet50}}^{\text{ADAM}} \Bigg). \tag{4.3}$$

The above score gives a single metric to measure the robustness of each model and optimization for all the noise settings. As seen from Equation 4.3, we use ResNet-50 with ADAM optimizer as a baseline setting.

Vulnerabilities due to training distribution shifts

The perturbed training samples corrupted by the learned input noise are shown in Figure 4-6. Perturbed training images for each class are presented in each column. All



(a) Clean (b) $N_p = 1$ (c) $N_p = 2$ (d) $N_p = 5$

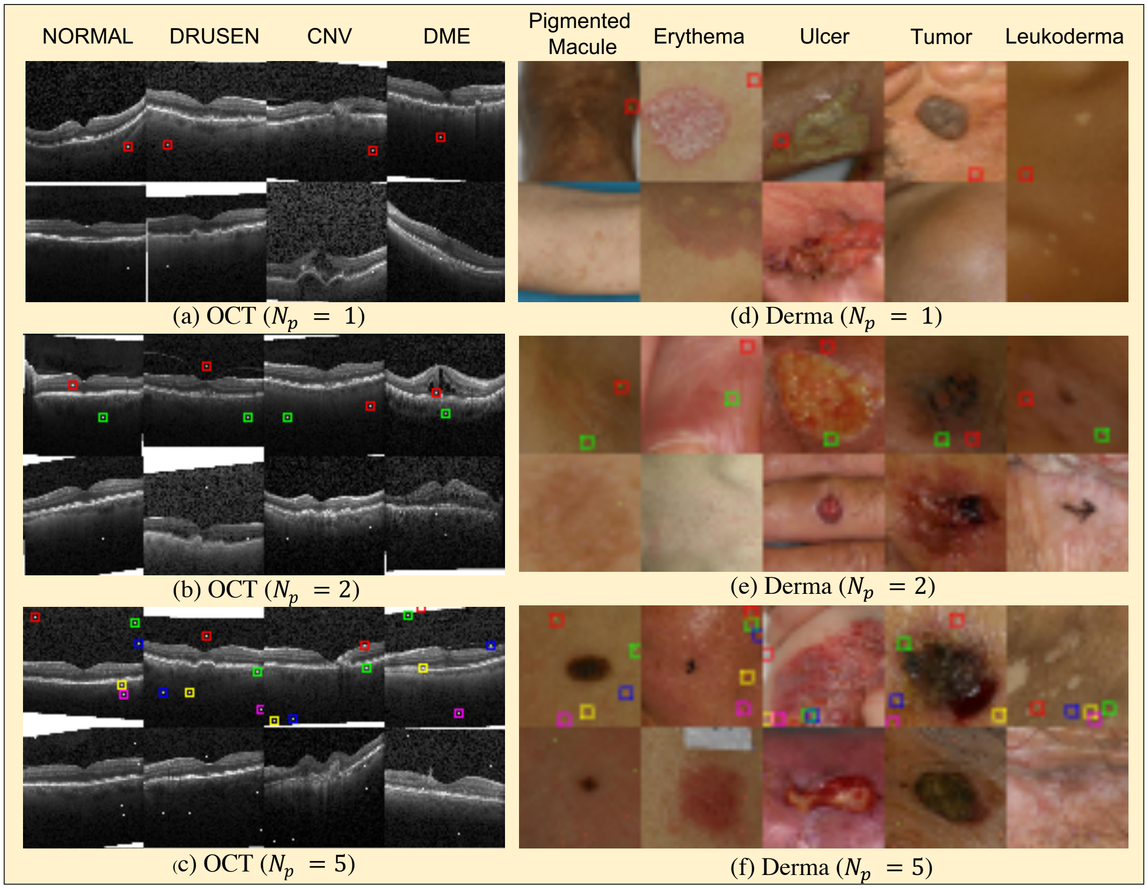
(e) Clean (f) $N_p = 1$ (g) $N_p = 2$ (h) $N_p = 5$

(i) Clean (j) $N_p = 1$ (k) $N_p = 2$ (l) $N_p = 5$

(m) Clean (n) $N_p = 1$ (o) $N_p = 2$ (p) $N_p = 5$

Figure 4-5: Showing Confusion matrix on clean testing images for DenseNet-121 on OCT dataset for ADAM (top row), Adabound (2^{nd} row), SGD (3^{rd} row), SGD-M (4^{th} row) for clean and corrupted training images. SGD based methods show better robustness compared to adaptive methods.

images belonging to that class have the same spatial distribution of pixel noise for each setting of N_p. Table 4-2 (a) provides the testing error for various models. For the OCT dataset, while learning from clean images exhibit training error close to 0.01%, simply adding a single-pixel adversarial noise in training increases error on clean images to 60% showing the vulnerability of deep models to training perturbations. A similar drop in performance can also be seen for the Dermatological dataset. Figure 4-2 (b) shows the learning curve for the CMA-ES based noise optimization algorithm which increases the fitness score with increasing generations. Higher pixel noise already starts at a high fitness whereas lower pixel noise is gradually optimized to find the best spatial location of corruption with the corresponding rise in fitness. Table 1



(a) OCT ($N_p = 1$)

(d) Derma ($N_p = 1$)

(b) OCT ($N_p = 2$)

(e) Derma ($N_p = 2$)

(c) OCT ($N_p = 5$)

(f) Derma ($N_p = 5$)

Figure 4-6: Perturbed training images from learned noise. Deep models trained on such pixel perturbed samples show very low accuracy on clean samples. Learned pixel noise is highlighted for top images in each block to ensure better visibility.

shows that precision, recall, and F1-score for each class significantly reduces for some classes which bring down the overall accuracy of the model on clean samples. All train accuracies are close to 100%.

SGD vs Adaptive Optimization

We also benchmark the generalization robustness of input normalization (CSN) and different optimization methods in the presence of pixel corruption in training images. Figure 4-3 shows the evolution of testing accuracy with increasing training epochs for various optimization methods. For both ResNet-50 and DenseNet-121, SGD-based methods (SGD and SGD-M) have higher testing accuracy compared to adaptive methods like ADAM and AdaBound. Train accuracies for SGD are around 60-70%.

Figure 4-4 measures the robustness of model-optimization pairs using our proposed robustness score in Equation 4.3. Input normalization shows enhanced robustness performance compared to the non-normalized settings, especially for SGD-based methods. For both ResNet and DenseNet models, adaptive optimization methods show poor generalization robustness compared to SGD. [105] explains this by reasoning that adaptive methods adjust the algorithm to the geometry of the data in an unconstrained fashion, hence they easily overfit to highly predictive features. On the other hand, SGD's optimization strategy is agnostic of the data manifold, only depending on the l_2 geometry inherent to learnable parameter space, thus exhibiting better generalization robustness. Figure 4-5 shows the confusion matrix for various optimizers.

4.4 Conclusions

In this work, we present the first benchmarking study regarding the generalization robustness of deep learning models on medical images under adversarial training perturbations. Such perturbations can possibly occur in practice due to device acquisition noise when training and testing images are collected from different sources. To study the *worst-case* performance, we generate optimal pixel perturbations using evolutionary strategy and examine various settings of learning architectures and optimizations. Our evaluations reveal inherent vulnerabilities in deep learning models, such that networks, when trained on mildly perturbed images, show significantly low testing accuracy on clean images. Furthermore, we perform analysis of different optimization techniques using our proposed noise aggregated robustness metric which shows that vanilla SGD based methods surprisingly exhibit much better generalization robustness compared to popular adaptive gradient-based methods. We hope that this study will inform researchers and practitioners in the medical image analysis field to benchmark their models on both SGD and adaptive gradient-based learning for improved generalization performance and drive more research on studying training time adversarial noise.

Chapter 5

Adversarial Robustness of Convolutional Models Learned in the Frequency Domain

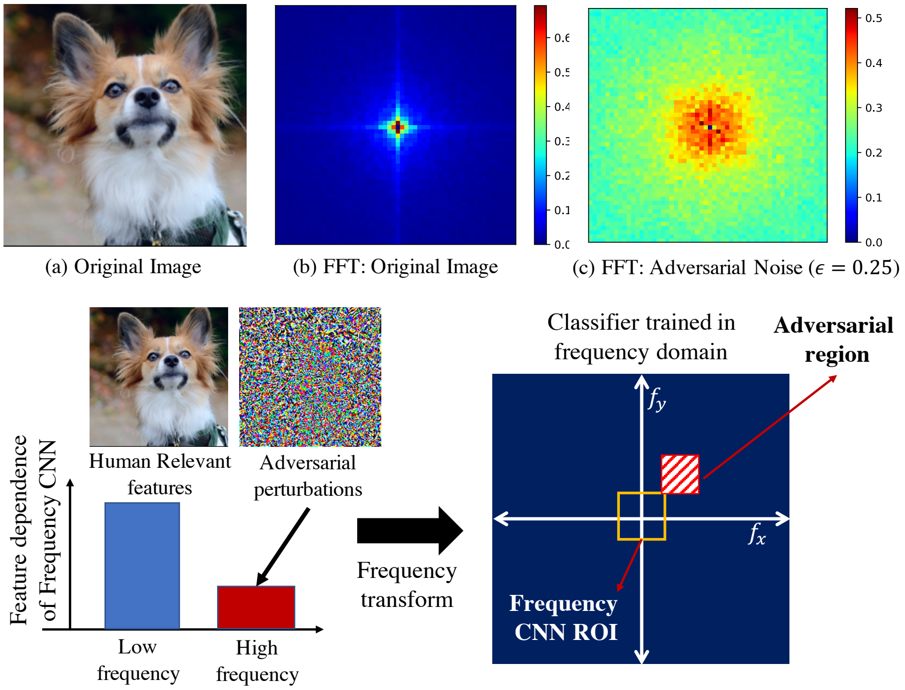
Recent works have extensively studied the robustness of standard CNNs for a variety of input noise, showing that such models exhibit vulnerabilities in the presence of small adversarial noise in the RGB input space. However, there has not been extensive robustness analysis of neural networks learned on frequency domain inputs. This work presents extensive comparisons of noise robustness between standard CNNs trained on image inputs and those trained in the frequency domain. We hypothesize that frequency domain learning of convolutional models confers the property to disentangle frequencies corresponding to semantic and adversarial features, thus resulting in adversarial robustness. Our experiments show that CNNs trained on Discrete Cosine Transform (DCT) inputs exhibit significantly better noise robustness to both adversarial and common spatial transformations compared to standard CNNs learned on RGB/Grayscale input. Our experimental evidence suggests that exploring frequency domain learning is a potential area to improve neural network robustness to test-time noise, thus warranting further research in this direction.

5.1 Introduction

Recent works in Convolutional Neural Networks (CNNs) [42,46,85,93] have showcased their immense academic and commercial contribution due to their notable empirical success in various application areas. CNNs have the advantage of automatically extracting suitable features from images which is a primary reason that such models outperform traditional computer vision systems. Added to that, the availability of large data in the age of the World Wide Web (WWW) has also contributed to the success of such deep learning systems. These models are utilized in various commercial applications that users engage in their daily life such as text recognition, facial recognition, object detection, etc. However, a recent line of work in adversarial examples [6, 16, 26, 56, 63, 98] have shown that small imperceptible noise in the CNN input can drastically fool the model to output wrong classification results with high confidence. Therefore, with such high stakes involved, we must ensure the safety and security associated with deep learning systems especially for critical applications such as autonomous driving or medical images analysis.

Neural networks are over-parameterized models. They have numerous trainable parameters (often millions or billions) from a very small amount of training data. This leads to cases where the model might not generalize to image inputs that are slightly different from naturally occurring images. Adversarial examples take advantage of this weakness to add small perturbations in the input image causing significant misclassification in such deep learning-based methods. Traditional adversarial attack methods [16, 21, 26, 35, 56, 63, 68, 75, 89, 98] fool trained neural networks during inference. These attacks add small imperceptible perturbations to the query images resulting in the classification function to cross the true class's decision boundary causing incorrect classification. Since these attacks are not easy to detect they can cause serious threats in real-world use cases such as object detection or classification applied to the autonomous driving case. This necessitates inventing robust methods and defenses against these malicious attacks to ensure safe commercial applications.

Previous works have mainly investigated the performance of neural networks in



(a) Original Image (b) FFT: Original Image (c) FFT: Adversarial Noise ($\epsilon = 0.25$)

(d) Frequency disentanglement in Fourier domain for defense against adversarial attacks

Figure 5-1: Our proposed concept of using frequency-based learning for adversarial defense. (a) The original image, (b) FFT on original image shows most information located on the central region of the image (average of 50 images), (c) FFT on difference image between adversarial and clean image shows that the attacked frequency components are more spread out compared to clean image semantic features, (d) We conceptualize that adversarial examples occupy frequency distribution in the higher frequency range whereas human-relevant features such as shape and size occupy low-frequency ranges. Thus, a CNN classifier trained in the frequency domain will learn to disentangle semantic and adversarial features in the frequency domain input resulting in robust performance.

the image domain. Adversarial attacks manifest as small perturbations that are not usually found in the natural distribution of the images. Therefore, such small changes can cause misclassification during inference because the perturbed image sample, although similar to human vision, was not witnessed during training. However, in this work, we perform a frequency domain analysis of adversarial examples. We argue with empirical evidence that adversarial features occupy frequency regions that are different from the semantic features that humans rely on for image classification. Typically, the frequency domain distribution of semantically useful features in natural images tends to be in the low-frequency region, whereas their adversarial counterparts

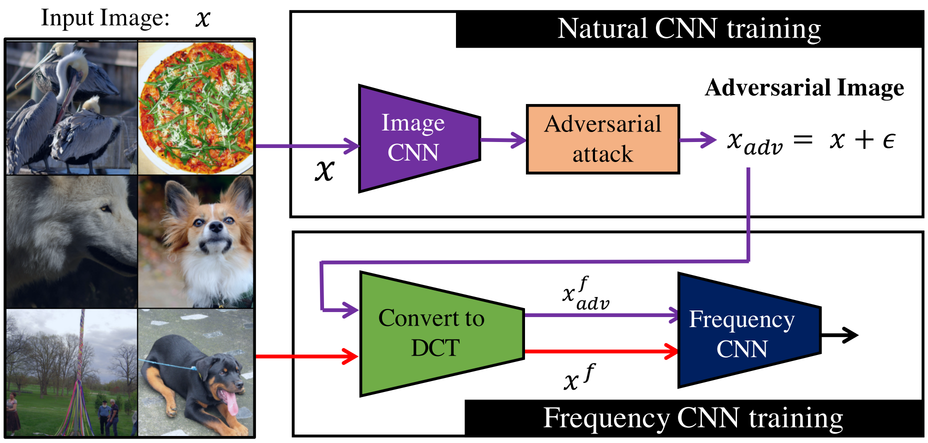


Figure 5-2: Showing the overview of adversarial defense in the frequency domain. We train both image and frequency domain CNNs with natural (x) and frequency-transformed (x^f) versions of the clean images. The frequency transformed adversarial image x^f_{adv} is passed through the frequency CNN to yield adversarial robustness. Red arrow shows training pipeline and purple arrow shows inference on perturbed images using the frequency CNN for adversarial robustness.

occupy higher frequency regions. Figure 5-1 illustrates the concept of disentangling the frequency occupancy regions by semantic and adversarial features. Therefore, we hypothesize that CNNs trained on the frequency domain inputs would learn to spatially focus on the semantically-relevant frequency components leading to better robustness against adversarial attacks that occupy a different frequency region.

In this work, we propose a defense method against adversarial attacks by learning CNN parameters in the frequency domain. Specifically, we transform the input image to YCbCr channels and compute their Discrete Cosine Transform (DCT), on which a CNN is learned via cross-entropy loss minimization. In parallel, we also train a CNN model on the RGB inputs. To analyze model robustness, we generate adversarial attacks (white box) from the RGB models. We analyze the accuracy of models under this setting and find that DCT transformation diffuses the attack features to frequency spectra that are not of interest to the CNN learned in the frequency domain for classification decision-making. Therefore, models trained on DCT inputs are more robust to adversarial perturbations.

Our method involves computing the DCT of the input image using the frequency domain learning outlined in [108]. We divide the image into smaller tiles and then perform DCT on each such tile (similar to JPEG compression). The computed DCT signal from the image is rearranged into multiple input channels and fed to the CNN, which is a modified version of ResNet [42], to handle multi-channel input. During inference, we input white box adversarial images to this CNN trained in the frequency domain. We show that the model trained on the frequency domain provides much better accuracy in the presence of adversarial noise when compared to the model learned on natural images. We validated our claims for multiple datasets and various attack strengths of three well-known adversarial attack methods. Furthermore, we demonstrate that CNNs learned in the frequency domain exhibit improved robustness to spatial transformations such as *worst-case* translation and rotations when compared to image domain CNNs.

Therefore, our contribution in this work can be summarized as

- We show that adversarial features occupy a separate region in the frequency spectrum that can be disentangled from the regions occupied by semantically meaningful features in natural images. We use this concept to propose an adversarial defense against popular adversarial attacks.

- We empirically show that learning in the frequency domain can be used as a defense against adversarial images by a feed-forward operation of the frequency domain transformation of the input adversarial image through the frequency CNN. This method of defense outperforms previous input transformation based adversarial defense methods.

- Finally, we show that our method is robust against spatial transformation attacks such as rotations and translations, to which naturally trained CNNs show poor performance as shown by the work of [28].

The rest of the chapter is described as follows: Section 2 describes the related works, Section 3 explains the overview of our frequency CNN-based defense against adversarial attacks. We provide an analysis of robustness against spatial transformations in Section 4. We present an extensive empirical evaluation of our method in

Section 5 for adversarial and spatial transformation and finally provide our conclusions in Section 6.

5.2 Related Works

In this section, we outline some previous works related to our method. We describe some works for adversarial and poisoning attacks, followed by some defensive measures against such attacks, and finally discuss some previous works that explore frequency domain analysis for adversarial robustness.

Adversarial Attacks: This recent line of work [6,16,26,56,63,98] demonstrated that it is possible to fool trained neural networks using *adversarial* query images that are imperceptible from normal unperturbed images. Su et al. [97] showed that it is possible to craft adversarial test images by single-pixel perturbations in training images. These attacks fall under the category of *evasive* attacks that exploit the weakness in trained models by attacking query images. In another kind of attack called data poisoning, the attacker injects malicious samples in the training data distribution to control the model behavior during test time. Such an attack was first introduced in the context of Support Vector Machines (SVM) for binary classification problems in [12]. Recently, there have been some works in the field of neural networks [96] as well. Koh et al. [55] used influence functions to synthesize adversarial training examples that can flip the predicted labels of a set of testing images. Shafahi et al. [91] used a forward-backward-splitting iterative procedure [32] to create targeted data poisoning attacks that performed better than previous methods. [19] proposed an evolutionary strategy based adversarial training time attack that caused overfitting in neural networks, leading to significantly poor out-of-sample performance.

Adversarial Defense: We review some generative learning methods for adversarial defense. [94] proposed a generative model called PixelDefend to detect adversarial samples and moving it back to the training data distribution. [64] used auto-encoders to detect adversarial inputs by using the reconstruction threshold and proposed a mechanism to defend against gray-box attack. [4] showed that learning with Variational Information Bottleneck (VIB) is robust to standard perturbation based adversarial example. [76] used knowledge distillation as a method to smoothen the decision boundaries which achieved adversarial robustness. Input transformation-based methods change the input adversarial image to weaken the strength of the noise. [109] used input transformations such as non-local smoothing, median smoothing, etc. Random input transformations such as cropping and resizing were proposed by [107] to ward off adversarial attacks. [80] proposed pixel deflection, a method to randomly shuffle pixels to improve robustness against adversarial attacks. In contrast to previous input transformation-based methods, this work shows a novel method of adversarial defense by learning a CNN model on DCT image inputs.

Frequency based Analysis: Frequency-based analysis of neural networks was carried out in recent research to understand the sensitivity of neural networks to different frequency components of attacked noise. [111] studies which how CNNs react to different noise basis vectors and show that the AutoAugment method of data augmentation is proved to be an effective method for improving model robustness. [110] explains the training behavior of neural networks in the frequency domain. [83] performs a frequency domain analysis in detail about stages of training in DNNs.

5.3 Proposed Method

In this section, we describe our method of learning in the frequency domain and how it can be used for adversarial defense. First, we explain how the image domain input is converted to the frequency domain. Next, we present how CNN trained in the frequency domain is robust to adversarial attacks.

5.3.1 Frequency Domain Signal

The image domain input consists of semantic features such as shape and color which are naturally identifiable by humans. However, adversarial attacks manifest in the natural domain and can cause misclassification for naturally trained CNN. For this purpose, we propose training a CNN in the frequency domain. Specifically, we choose the learning in the frequency architecture proposed by [108].

We found that naively converting an image domain input to the DCT domain causes a drop in the accuracy on the test set. The reason for this might be because the frequency transformation limits the entire information of the image in a spatially local frequency region as shown in Figure 5-1(b). A desirable property of the frequency domain CNN should be high training accuracy which is at-par with the CNN trained on natural images. For this purpose, we found that the “learning in the frequency domain” work from [108] is appropriate to obtain high test accuracy while still learning in the frequency domain.

In this method, we transform the input image to YCbCr colorspace. For each channel in the converted image, we compute the Discrete Cosine Transform (DCT). Next, we join the frequency components belonging to the same frequency group to obtain a multi-channel input. This rearrangement of the frequency channels to multiple inputs is the primary reason for improving the training accuracy. In such a multi-channel case, the information about local patches in the image is distributed across all channels, thus resulting in better feature extraction that entire image frequency transformation. In our case, we set the number of input channels to be 24 following the original paper. For training the model corresponding to this multi-channel input, we use the modified ResNet model as proposed by [108], which produces accuracy metrics comparable with its image domain counterpart CNN model. For experiments with the ImageNet dataset, we use the above frequency transformation. However, for experiments with Fashion-MNIST and SVHN datasets, we use direct DCT on the entire image because we found that it produces comparable accuracies in both domains.

5.3.2 Adversarial Defense in the Frequency Domain

Consider we are given an dataset consisting of samples, $(\boldsymbol{x}, y) \sim \mathcal{D}$ from the true data distribution D and $\boldsymbol{x}$ is an instance of image and y the corresponding label. We are interested in learning a classification task. For this purpose, we learn a CNN model given as $F_{\text{img}}(\boldsymbol{x}; \boldsymbol{\theta})$ parameterized by $\boldsymbol{\theta}$. The image domain model is learned using standard cross-entropy minimization

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\big[\mathcal{L}_{CE}(F_{\text{img}}(\boldsymbol{x};\boldsymbol{\theta}), y)\big], \tag{5.1}$$

where $\mathcal{L}_{CE}$ is the cross-entropy loss. We choose ResNet-50 [42] model for the CNN model in our experiments.

Similarly we define the corresponding frequency domain model, $F_{\text{dct}}(\boldsymbol{x};\boldsymbol{\theta}_{\text{dct}})$ which is trained using the multi-channel frequency domain signal as described in Section 5.3.1, using the following equation

$$\boldsymbol{\theta}^*_{\text{dct}} = \arg\min_{\boldsymbol{\theta}_{\text{dct}}} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\big[\mathcal{L}_{CE}(F_{\text{dct}}(\boldsymbol{x}^{\text{dct}};\boldsymbol{\theta}_{\text{dct}}), y)\big], \tag{5.2}$$

where $\boldsymbol{x}^{\text{dct}} = \mathcal{F}^{\text{dct}}(\boldsymbol{x})$ is the frequency domain transformed image. We use the implementation of JPEG filters for fast computation of the DCT transformed image.

We generate white-box adversarial images by finding a perturbation δ that maximizes the cross-entropy loss within a feasible search area. Here, we outline the method adopted by [63] for finding the adversarial perturbation within the feasible region Δ. The following optimization is performed to obtain the best perturbation, δ^*, given as

$$\delta^* = \arg\max_{\delta\in\Delta} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\big[\mathcal{L}_{CE}(F_{\text{img}}(\boldsymbol{x}+\delta;\boldsymbol{\theta}^*), y)\big], \tag{5.3}$$

where depending on the distance norm of the adversarial perturbation $(\ell_\infty, \ell_2, etc.)$ the feasible region Δ is decided.

The above-perturbed signal when given as input to the image domain CNN would result in very low test accuracy. To improve the accuracy of the model under such adversarial attacks, we apply defensive strategies such as [80, 109] to improve the performance. For defense strategy using our method, we feed-forward the DCT transformed adversarial image, as $y_{dct} = F_{\text{dct}}(\mathcal{F}^{\text{dct}}(\boldsymbol{x}+\delta);\boldsymbol{\theta}^*_{\text{dct}})$. We report the accuracy for this predicted softmax score and show that using this method the test accuracy is much better compared to the image domain defensive strategies. The superior performance is attributed to the frequency disentanglement of the adversarial noise features and

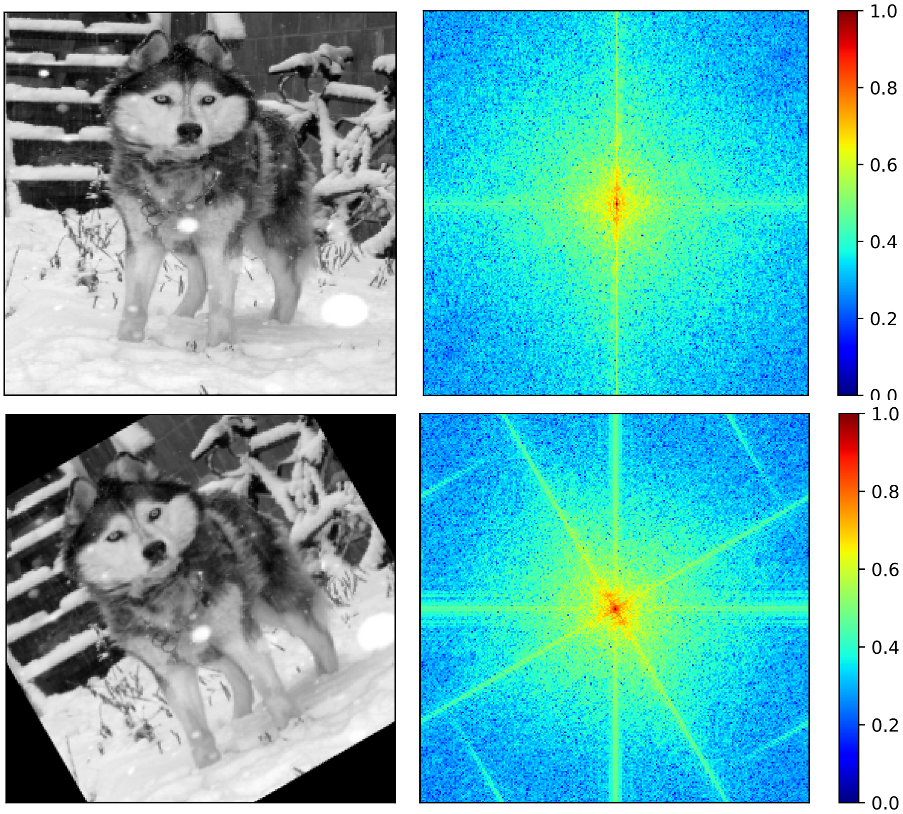


Figure 5-3: Showing that the FFT of a rotated image signal is also rotated in the frequency domain. Artifacts are also seen due to the image getting cropped.

the semantic features. Since F_{dct} focuses on the semantic frequency region, adversarial noise cannot reduce the accuracy due to separate frequency range occupancy for such additive perturbations. Figure 5-2 shows the overview of our adversarial defense using the frequency CNN method.

5.4 Spatial transformation

Although CNNs are claimed to have translational invariance due to pooling operations, previous works [28, 115] have shown that they are susceptible to spatial transformations. For example, a CNN model trained on naturally occurring images will produce low accuracy when tested on unseen shifts (or translations) and rotations. This is attributed to the fact that the features learned by CNNs are not invariant to the spatial transformations which produced the mismatch between the training distribution and test-time query samples, resulting in poor generalization.

We show that learning a CNN in the frequency domain can alleviate these problems due to the invariance of the learned frequency-domain signals. Instead of capturing semantic properties such as shape and size, frequency-domain models capture frequency contents of the image that are invariant to translations. Therefore, such learned features are robust and can sustain high test accuracy, even for out-of-sample images with unseen spatial transformations. Below, we discuss the cases of translation and rotation.

5.4.1 Translation Invariance of Magnitude Spectrum

Consider $I(x, y)$ as an input image having translation (a, b) in the x and y axis respectively. The final translated image is given as $I_t(x, y) = I(x - a, y - b)$. The Fourier transform of the translated signal amounts to the following

$$I_t^f(\Omega_x, \Omega_y) = \mathcal{F}(I_t) = e^{-(a\Omega_x + b\Omega_y)} I^f, \tag{5.4}$$

where $I^f = \mathcal{F}(I(x, y))$ and Ω_x, Ω_y are the continuous frequency variables in the x and y axis. This result shows that the magnitude spectrum for both the original and translated signals is the same with the only difference in the phase spectrum. Similar results are obtained in the discrete variable version of the input. From the above results, we hypothesize that the features learned in the DCT domain model are invariant to arbitrary shifts in the input image. We empirically verify our claims in the experimental results section 5.5.4 that exhibit better robustness to translations compared to image domain training.

5.4.2 Robustness Against Image Rotations

We hypothesize that the features learned by the CNN trained in the DCT domain are also robust to rotational transformations. Typically, we use rectilinear co-ordinates for the frequency domain transformation leading to equivariance for the rotational transformation, implying that rotation of the image also leads to rotation of the frequency signal as shown in Figure 5-3. However, we hypothesize that since most of the discriminative features learned by the DCT CNN is concentrated in the low-frequency region where the feature rotation is small, feature variance under rotation is also small. Furthermore, in our case we perform, frequency transformation in a block-wise manner as described in Section 5.3.1. Therefore, the locally relevant features in each channel of the frequency signal show very small variance from the original non-rotated signal leading to better performance of frequency domain CNN model to unseen rotational transformations during inference. We empirically substantiate this claim in the experimental results section.

5.5 Experimental Results

In this section, we provide empirical results of our experiments. Specifically, we are interested in knowing the answer to the following questions: (1) Does learning in the frequency domain improve the adversarial robustness of the models to natural image domain adversarial attacks? (2) Does frequency domain learning render spatial transformation attacks useless due to invariance under such attacks?

5.5.1 Setup

We use ResNet-50 models for our natural image CNN. For the frequency domain, we use the ResNet version of the frequency CNN with the number of channels 24. We report the results on ImageNet, Fashion-MNIST, and SVHN datasets. For the ImageNet dataset, we choose 2000 images from the validation set such that they produce 100% accuracy for both the natural image trained and frequency trained CNN. For the Fashion-MNIST dataset, we choose 5000 randomly chosen images from the test set for both the adversarial and spatial transformation noise experiments. For the SVHN dataset, 2000 randomly chosen images from the test set were used for adversarial robustness experiments. In the case of the ImageNet dataset, we computed the adversarial images and saved them to the disk. During inference, the images were read from disk. Thus, some quantization might have occurred in

Table 5.1: Quantitative comparison of various input transformation based adversarial defense methods in the image domain with our proposed defense in the frequency domain for ImageNet dataset. We show top-5 accuracy in this table for various adversarial attack methods. Our DCT-based learning method consistently outperforms previous image defensive strategies for both the noise attack strengths of $\epsilon = 0.15$ and $\epsilon = 0.25$. Top-1 accuracy scores are shown in the supplementary materials.

	PGD [63]		BIA [56]		Momentum [26]	
	$\epsilon = 0.1\ 5$	ϵ= 0.25	$\epsilon = 0.15$	ϵ= 0.25	$\epsilon = 0.15$	ϵ= 0.25
Clean Image	100.0	100.0	100.0	100.0	100.0	100.0
No Defense	50.9	35.5	36.0	15.5	26.6	14.4
Bitwise [109]	50.9	35.6	36.1	15.5	27.0	14.4
Median Smoothing [109]	82.9	78.9	79.7	70.0	70.25	56.4
Average Smoothing [109]	88.3	84.7	86.2	78.7	79.9	69.5
PixelDeflect (PD) [80]	80.3	70.2	74.9	54.1	67.4	44.4
PD [80]+Bitwise [109]	80.4	69.5	75.6	55.7	68.3	45.4
PD [80]+Median Smoothing [109]	84.5	80.2	80.3	71.3	72.7	60.4
PD [80]+Average Smoothing [109]	87.9	84.2	86.3	80.0	81.0	72.5
Ours (DCT)	**95.4**	**94.0**	**93.7**	**88.0**	**89.9**	**79.1**

the image compression process. For the other two datasets, DCT of the adversarial images were performed in memory and no such compression artifacts were present.

We explain the details of our experiments in this section. Specifically, we discuss the details of the adversarial attacks, defensive strategies, and pre-processing for frequency CNN and Image-CNN learning.

Adversarial Attacks

We use three adversarial attacks for testing the robustness of our systems: (i) Projected Gradient Descent (PGD) [63], (ii) Basic Iterative Attack (BIA) [56], and (iii) Momentum-boosted Attacks [26]. For the implementation of these attacks, we used Advertorch library [24] for finding white-box attacks. The respective attack class that we use for the attacks are (i) `LinfPGDAttack` for PGD attack with $n_{iter} = 100$, (ii) `LinfBasicIterativeAttack` for BIA with $n_{iter} = 20$, and (iii) `LinfMomentumIterativeAttack` for Momentum Attacks with $n_{iter} = 100$, where n_{iter} being the number of iterations to find the adversarial perturbation.

Adversarial Defense

We use the defensive strategies outlined in [109] specifically the following input transformation defenses, (i) BitSqueezing with bit width=5, (ii) MedianSmoothing with kernel size=5, and (iii) AverageSmoothing with kernel size=5. Additionally, we add PixelDeflection attacks [80] for improving the overall input transformation defensive performance. For PixelDeflection we use $\sigma = 0.04$, number of pixel deflections=10,000 and window size=40.

Pre-processing for RGB and DCT DataLoaders

We define the data augmentation and pre-processing for the image and DCT based CNN learning as follows:

Image input: For the Fashion-MNIST and SVHN datasets, we do not use data augmentations for training the CNN. In both cases, the input images were normalized within the range $[-1, 1]$. For the ImageNet dataset, we resize each image in the dataset to a size of (256×256) and then perform the center crop of size (224×224). The normal mean and standard deviation values used in [42] is employed for normalizing the images.

The above data loader (in PyTorch) is used for obtaining clean images. For computing the performance on adversarially attacked images, we first compute the attacked image as PyTorch tensor in memory using the methods described in Section 5.5.1. Next, we save the images on the disk as JPEG images. For computing the performance of our various CNN methods, we use a data loader on these saved images without any data augmentation. The JPEG encoding process might introduce some artifacts that might impart some defense against the adversarial attacks.

DCT input: For Fashion-MNIST and SVHN datasets, we compute the Discrete Cosine Transform (DCT) for the entire image using OpenCV library [73]. For the ImageNet dataset, we use the method outlined in [108]. Before DCT computation, the input images are first resized to (512×512) and then center cropped to size (448×448). After this step, we compute the YCbCr and divide the images into small

Table 5.2: Quantitative comparison and ablation study of various input transformation based adversarial defense methods in the image domain with our proposed defense in the frequency domain for ImageNet dataset. We show top-1 accuracy in this table for various adversarial attack methods. Our DCT-based learning method consistently outperforms previous image defensive strategies for both the noise attack strengths of $\epsilon = 0.15$ and $\epsilon = 0.25$. Top performing test accuracy is shown in **bold** and second best performance is shown in red color.

	PGD [63]		BIA [56]		Momentum [26]	
	$\epsilon = 0.1\ 5$	$\epsilon = 0.25$	$\epsilon = 0.15$	$\epsilon = 0.25$	$\epsilon = 0.15$	$\epsilon = 0.25$
Clean Image	100.0	100.0	100.0	100.0	100.0	100.0
No Defense	03.0	01.0	02.5	13.2	00.4	00.1
Bitwise [109]	03.0	01.1	02.5	00.1	00.4	00.1
Median Smoothing [109]	47.3	42.7	44.0	29.1	30.0	17.8
Average Smoothing [109]	58.1	51.0	56.8	41.1	43.4	28.6
PixelDeflect (PD) [80]	40.5	23.4	36.2	13.2	24.0	06.1
PD [80]+Bitwise [109]	41.0	24.1	36.0	12.8	24.1	6.7
PD [80]+Median Smoothing [109]	50.5	44.2	46.8	31.4	31.3	19.8
PD [80]+Average Smoothing [109]	59.1	53.3	57.6	46.0	48.7	36.1
Ours (no-center-crop+DCT)	72.7	67.7	69.2	52.3	54.4	38.8
Ours (center-crop+DCT)	**72.9**	**68.0**	**69.8**	**54.7**	**57.5**	**43.1**

8×8 tiles and compute the DCT on each such tile. To obtain this process, we used the TurboJPEG library [1]. The same process is applied to the adversarial images that are read from the disk.

Spatial Transformation

For RGB input, the affine transformation module used in [28] was used for computing the image transformation on the torch tensor directly. For the DCT image, we used `cv2.warpAffine` operation from the OpenCV library [73] for computing the spatial image transformations. The DCT computation operations that we described above are then performed after the spatial transformations.

5.5.2 Robustness to Adversarial Attacks

We show here that our hypothesis of adversarial robustness for learning in the frequency domain is valid for multiple datasets.

ImageNet: Table 5.1 shows the testing accuracy of the ResNet-50 model and

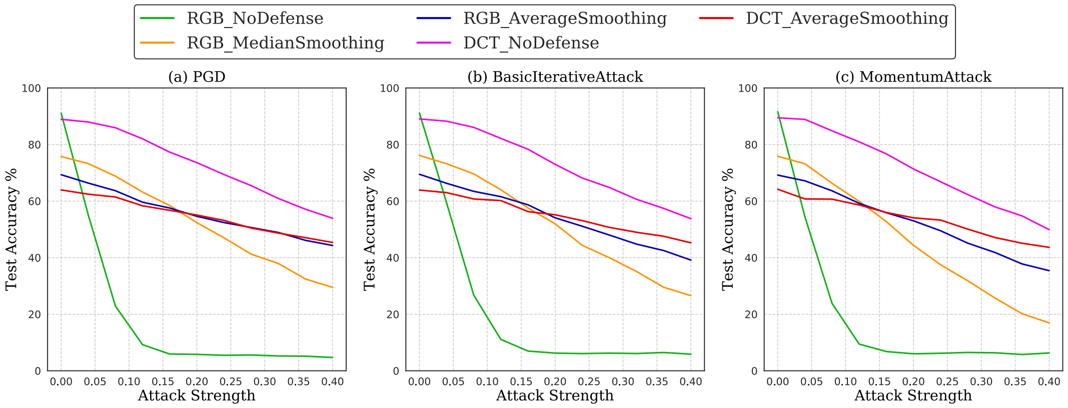


Figure 5-4: Performance of our DCT CNN models compared to other defense methods on Fashion-MNIST test set. Our proposed defense using DCT-based training is robust compared to previous defense methods. The performance does not drop significantly even with the increasing strength of adversarial attacks.

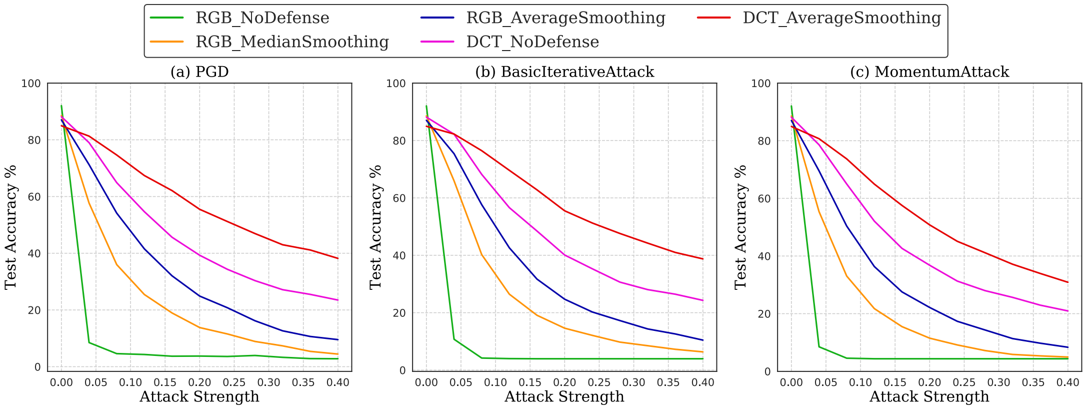


Figure 5-5: Performance of our DCT CNN models compared to other defense methods on SVHN test set for various attack methods and attack strengths.

ResNet-24 channel DCT model. We compare with previous input transformation methods from [109] and also couple the pixel deflection defensive strategy [80] with these methods. Even with multiple defensive strategies, our method outperforms PD+Average Smoothing, which is the best performing input image transformation based defense.

Fashion-MNIST: Figure 5-4 shows the performance of frequency domain CNN compared to previous input transformation based defenses. Our DCT domain CNN shows better robustness compared to all previous methods. Average Smoothing exhibits the best performance in all the previous defenses which is in line with our

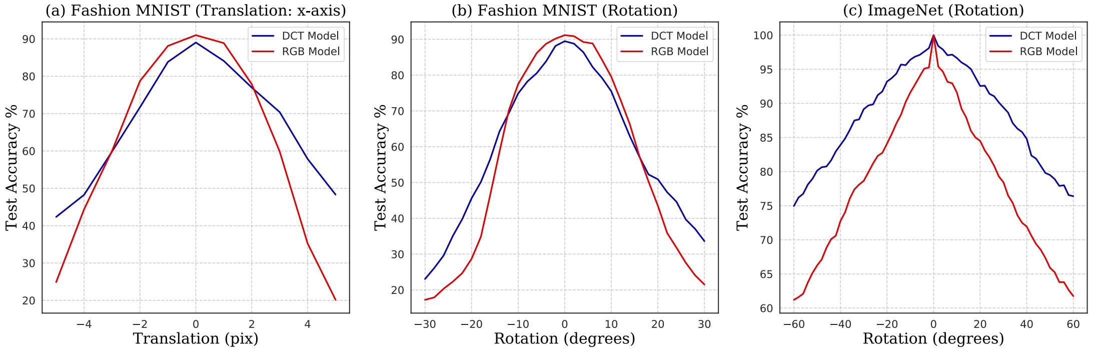


Figure 5-6: Our DCT CNN model shows better robustness to spatial transformations such as translation and rotation compared to image-based CNN models as evidenced by higher testing accuracy , especially on the extreme cases, for the following three scenarios: (a) Translation in the x-axis for Fashion-MNIST dataset, (b) Rotation on the Fashion-MNIST dataset, and (c) Rotation on the ImageNet dataset.

assumption that semantically meaningful features occupy a low-frequency range. Our DCT-based learning method with no additional input transformation performs the best. The addition of Average Smoothing in addition to DCT-based learning reduces overall accuracy but it still outperforms image domain-based defenses for high noise strength.

SVHN: Figure 5-5 shows the behavior of DCT CNN that exhibits superior adversarial robustness compared to naturally trained models even with adversarial defense. In the RGB defense strategies, Average Smoothing performs the best. However, our DCT+Average Smoothing method outperforms this method especially for high attack strengths of the adversarial attacks.

5.5.3 Ablation Study: ImageNet Adversarial Defense

The robustness properties of our method on the ImageNet dataset are presented in Table 5.2 showing top-1 accuracy for the three adversarial attacks. The corresponding top-5 accuracies are shown in the main text. For all the cases, the CNN trained on the DCT input consistently outperforms the testing accuracy on RGB trained CNN on the adversarial attacked images.

Since the DCT computation involves resize and cropping operations that are shown to impart adversarial defense by [107]. Therefore, we performed ablation stud-

Table 5.3: Test accuracy under spatial transformation of Image CNN vs DCT CNN. For random case, we report the mean of 10 sampled translations or rotations values. Our method shows better performance for both datasets.

		Rand. T	Rand. R	Worst case T	Worst case R
Fashion-MNIST	Image CNN	59.8	54.3	20.1	17.2
	DCT CNN	**67.4**	**57.4**	**42.3**	**23.4**
ImageNet	Image CNN	91.8	78.0	78.9	61.2
	DCT CNN	**96.5**	**87.6**	**90.2**	**75.0**

ies where we removed the center-crop operation and directly performed DCT of the entire image. As seen in Table 5.2, even without the CENTER-CROP operation, DCT trained CNN outperforms previous CNN methods.

5.5.4 Robustness to Spatial Transformations

Here we show that the frequency domain CNN is more robust to spatial transformations such as translation and rotation, for various datasets. The work of [28] has shown that CNNs can be made to have low testing accuracy with spatial transformations to the input images. We show here that CNNs trained in the frequency domain shows better resilience to such spatial transformations.

For the computation of spatial transformations on the RGB inputs and DCT inputs for Fashion-MNIST, we applied the affine transformation module provided by [28] that directly performs a spatial transformation on PyTorch tensors. However, since DCT computation using [108] required NumPy tensor as input, spatial transformations for the DCT input in the case of ImageNet images were performed using OpenCV [73] image transformation operations.

Translation: Figure 5-6(a) and Table 5.3 shows the performance of both image domain and frequency domain CNNs to translations. Our DCT CNN consistently performs better than the image domain CNN showing better translational robustness for Fashion-MNIST and ImageNet dataset. For Fashion-MNIST results in Figure 5-6(a), the frequency CNN initially performs poorer than the image CNN model, however, at extreme translations, the DCT-based model shows better performance.

Rotation: We show that our model trained on the DCT image signal is robust to rotational transformations. We applied rotations from -30^o to $+30^o$ in gaps of 2^o for Fashion-MNIST dataset as shown in Figure 5-6(b). For ImageNet dataset, results for rotations from -60^o to $+60^o$ in gaps of 2^o are shown in Figure 5-6(c). Table 5.3 shows the randomly sampled and worst-case performance under rotational transformations. All the results show that our DCT CNN outperforms the image domain CNN is most cases of rotational transformation especially for the case where the rotational angle is high.

5.6 Conclusions

In this work, we present an alternative view for adversarial robustness using frequency domain transformations on image signals. Firstly, we leverage the fact that adversarial perturbations by common attack methods occupy frequency regions that do not coincide with the semantically meaningful features. Therefore, they can be disentangled in the frequency domain. We show that training a CNN model on the DCT of the input image, learns to focus on the semantically meaningful frequency range (typically low frequency) and thus adversarial attacks do not affect the DCT trained CNN due to frequency disentanglement. This property allows CNNs trained in the frequency domain to show better adversarial robustness. Furthermore, we demonstrate that the poor performance of CNNs trained in the image domain to spatial transformations can also be alleviated by training CNN on the frequency domain. Our results illustrate that training CNNs in the frequency domain provides additional robustness that is not obtained by training in the image domain. Therefore, further research is warranted in this direction for creating CNNs robust to malicious attacks, thus, making it applicable for commercial applications. Specifically, one direction for future research can be directly attacking the frequency domain model using gradient-free methods and leveraging the duality principle of frequency transformation to obtain robustness in the image domain.

Chapter 6

Conclusions

6.1 Summary of Thesis Contributions

In this thesis, we have presented a new kind of adversarial attack on training images that causes high generalization loss on clean test images. This exposes the vulnerability of CNNs to training time attacks. Specifically, we outline an evolutionary strategy based attack that corrupts automatically selected pixels in the training images that maximally reduces accuracy on clean test images. Using this attack method, we analyze the performance of neural network properties related to generalization and prevention of overfitting in the presence of regularization methods. We also analyze various loss functions and show that vanilla cross-entropy loss which is widely used in various deep learning classification tasks are not robust to training noise and hence leads to poor performance in the presence of training noise. We proposed an improved training loss that considers the mutual information between the learned features and the labels, which shows improved performance in the presence of training noise. Furthermore, we use our method to benchmark popular neural network optimization methods and show that SGD based methods are more robust compared to adaptive optimization methods in the presence of training noise. We further showed that our proposed attack methods can also attack security-critical applications like medical image classification etc, thus showing the impact of such training time noise attacks. Finally, we perform a frequency-domain analysis of the adversarially attacked images and show that the CNNs trained on frequency domain input shows better robustness to adversarial attacks. We hypothesize that adversarial noise and semantically useful features occupy different frequency range that can be disentangled by the CNN learned in the frequency domain thus enabling best adversarial robustness.

6.2 Direction of Future Works

While working on this thesis, I came up with various ideas that I did not have the time to implement. However, I would like to list these works here as possible future works for this thesis.

6.2.1 Extending Proposed Attacks on Object Detection and Semantic Segmentation

This thesis proposed the adversarial attack for classification models but adding single pixel noise in the training images in a class-wise fashion. Object detection methods like [31, 86] use classification in one of the stages of object detection pipeline which can be attacked using our method. The idea is to use few pixel attacks within the annotated bounding box of the objects that can cause loss of generalization in such models when tested on clean samples. Similarly, this method can be extended to semantic segmentation based methods [20, 61, 87] as well by adding pixel attacks on the different segments of the images. There can be different variations of the superimposed attacks which definitely requires effort in the future.

6.2.2 Training Time Attacks on Natural Language and Speech Classifiers

Deep Neural Networks (DNNs) have also been used in NLP applications [23, 82] and speech classification tasks [2] as well. However, works like [49] show that these models are vulnerable against adversarial attacks as well. Our proposed method for training time attack can also be extended to these modalities. In the case of natural language inputs, our method can be used to insert a distractor token in a class specific manner to the input training samples resulting in association of that token with that specific class. During classification on clean texts, due to missing entry of that specialized token, the text classification model would cause behavior close to random classifier. Similar attacks are also possible in case of speech classification by artificial insertion of special phonemes in the input speech signal.

6.2.3 Extending the Pixel Attack to General Attacks

In this thesis, we have limited the kind of attack for few pixel attacks on training images. However, our formulation of the attack allows general attack type that is not limited to pixel attacks but also can be applied to spread out attacks similar to the ones that is found in traditional adversarial attack methods like [16, 21, 26, 35, 56, 63, 68, 75, 89, 98]. The idea is to use a parameterized family of attacks that spreads the attack over the entire pixel space of the image using gradient-free evolutionary algorithms. Our method can also be extended to have differentiable functional forms that can be optimized using autodiff methods.

6.2.4 Attacking Imitation Learning Frameworks

While this thesis mainly focuses on attacking from supervised learning, this can also be used for attacking reinforcement learning methods, specifically imitation learning techniques that learn action policies by imitating an expert's trajectories. Our method can inject spurious data in the input expert trajectories that can direct the learned policy to follow a malicious target behavior designed by the attacker. Specifically, for discrete action environments, adding key-point attacked frames in the expert data might be able to deviate the policy from the true expert trajectories thereby causing unnatural behavior.

Appendix A

Additional Results for EvoShift algorithm

A.1 Perturbed training samples

The attacked training images after adding the optimally learned pixel perturbations are shown for MNIST (Figure A-1), Fashion-MNIST (Figure A-2), CIFAR10 (Figure A-3), and SVHN (Figure A-4) datasets below. These pixel attacks are corresponding to the best cost value from the *EvoShift* CMA-ES algorithm. We can observe from the figure that smaller pixel perturbations are difficult to identify for humans, especially for colored CIFAR10 and SVHN datasets. However, such small perturbations are enough to cause significant overfitting in neural networks as shown in the main paper experimental section.

A.2 Plots for all training and Test Cases

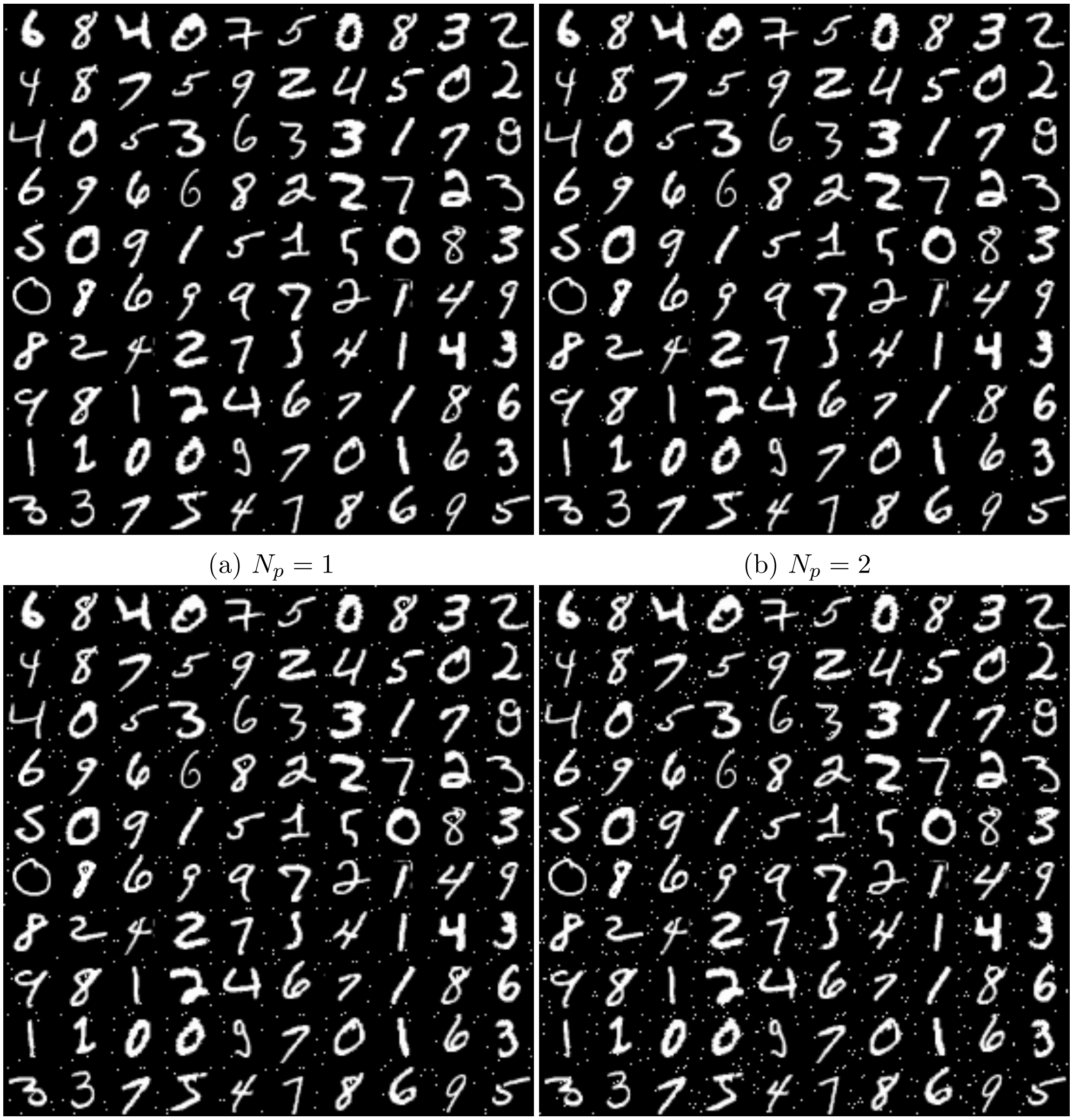
We elaborate the training and testing accuracy for the factors of (i) explicit regularization and (ii) optimization technique, for all the noise settings of 1, 2, 5, and 10-pixel perturbations.

Explicit regularization testing accuracy: Figure A-5 shows the testing accuracy for different regularization techniques for various datasets. Random crop data augmentation is the best regularization for $N_p = 1$ case. However, for higher pixel perturbations, $N_p = 2, 5, 10$, even data augmentation regularization is unable to produce high accuracy.

Explicit regularization training accuracy: Figure A-6 shows the corresponding training accuracy for different regularization techniques. For most cases, the training accuracy goes to 100% while the corresponding testing accuracy is low, showing strong signs of overfitting in the presence of our proposed attack.

Choice of Optimization Testing accuracy: Figure A-7 shows the testing accuracy for different optimizer. Stochastic Gradient Descent (SGD) based methods show better testing accuracy compared to adaptive optimization methods, showing better robustness to training perturbations.

Choice of Optimization Training accuracy: Figure A-8 shows the corresponding training accuracy for different optimization techniques. For most optimizers (except SGD), the training accuracy quickly goes to 100% showing overfitting in the presence of our proposed attack, especially for adaptive optimizers. SGD based methods show slow convergence.



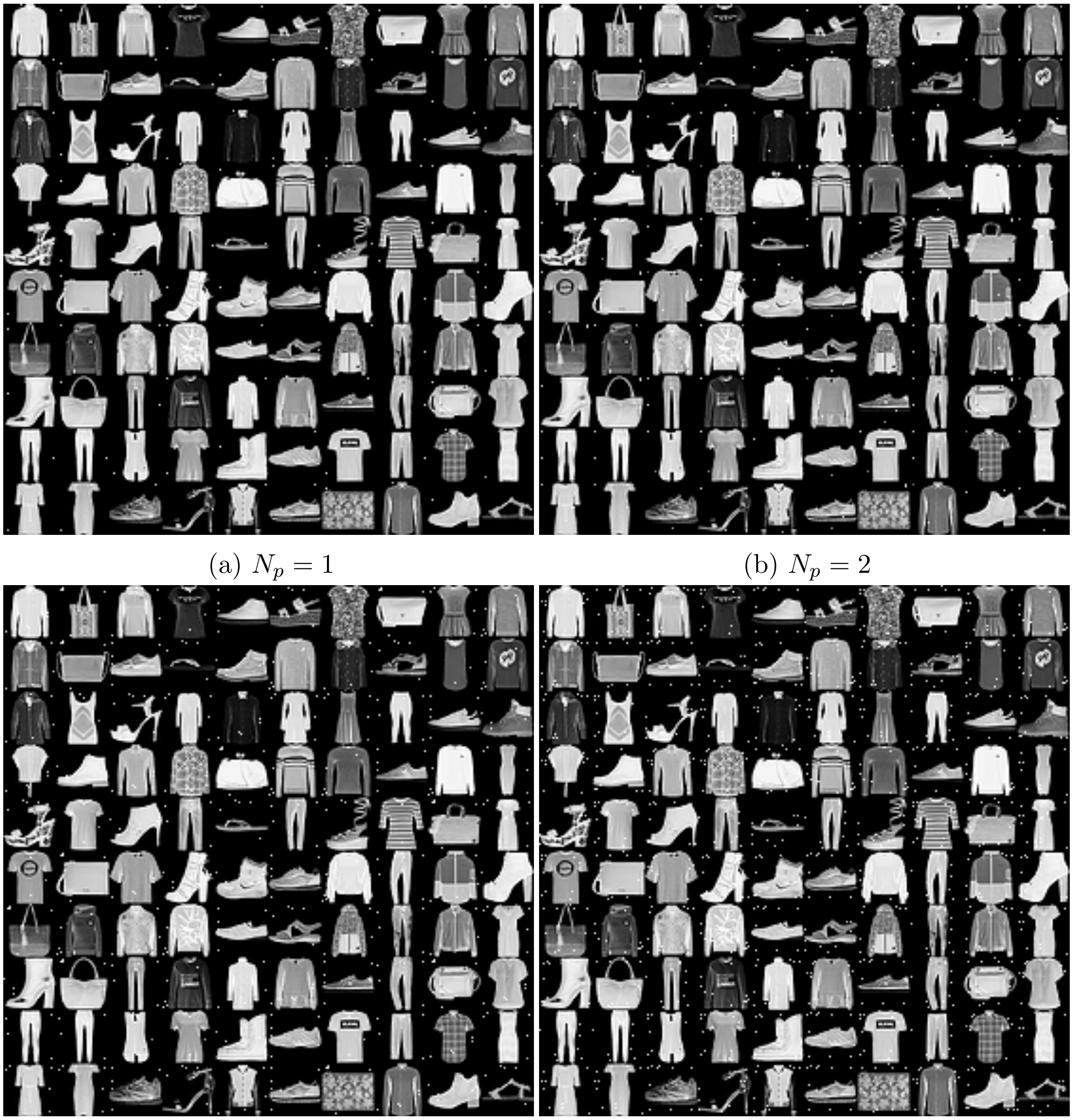
(a) $N_p = 1$

(b) $N_p = 2$

(c) $N_p = 5$

(d) $N_p = 10$

Figure A-1: Attacked training images for MNIST dataset.



(a) $N_p = 1$

(b) $N_p = 2$

(c) $N_p = 5$

(d) $N_p = 10$

Figure A-2: Attacked training images for Fashion-MNIST dataset.

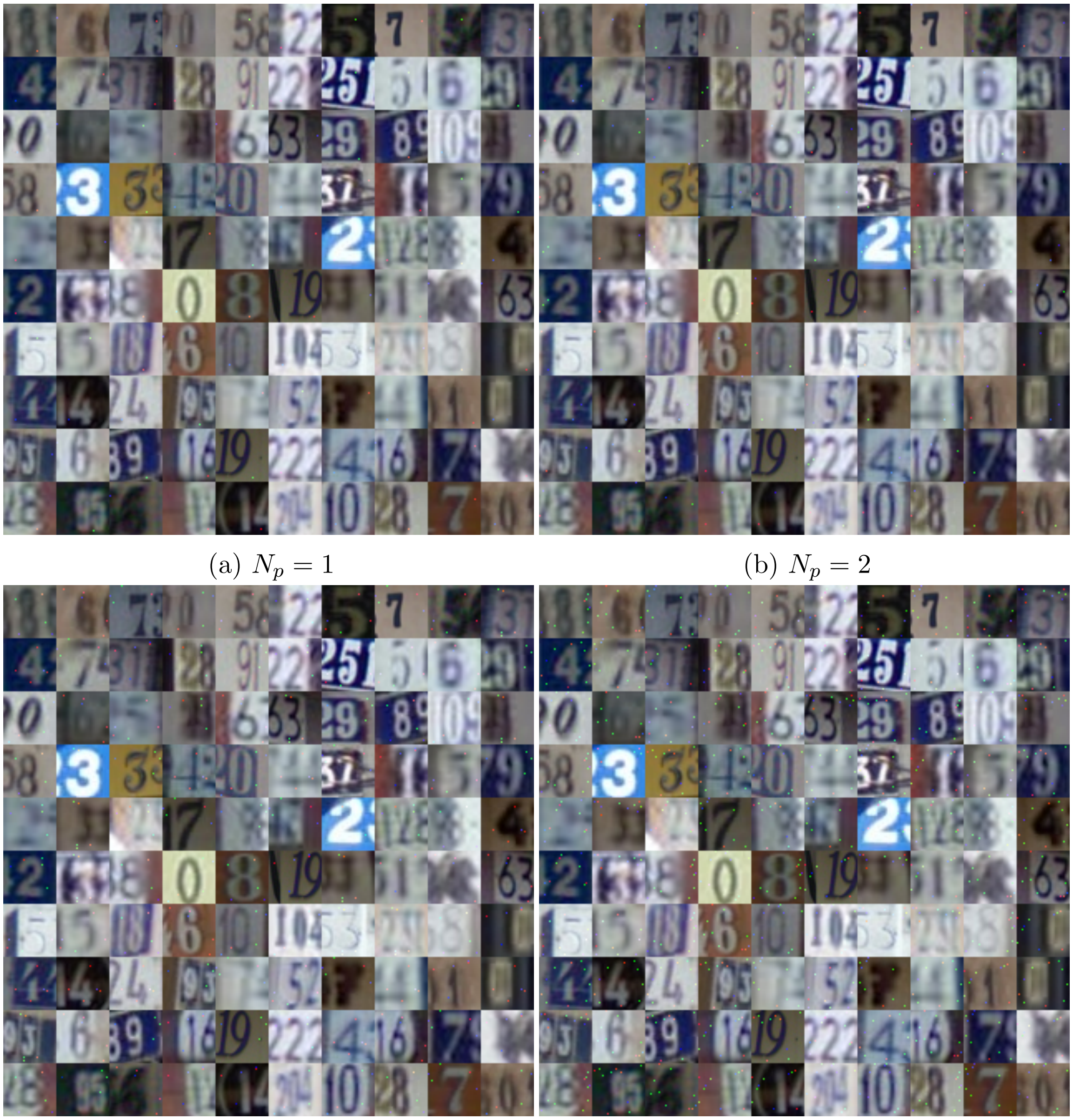
(a) $N_p = 1$

(b) $N_p = 2$

(c) $N_p = 5$

(d) $N_p = 10$

Figure A-3: Attacked training images for CIFAR10 dataset.



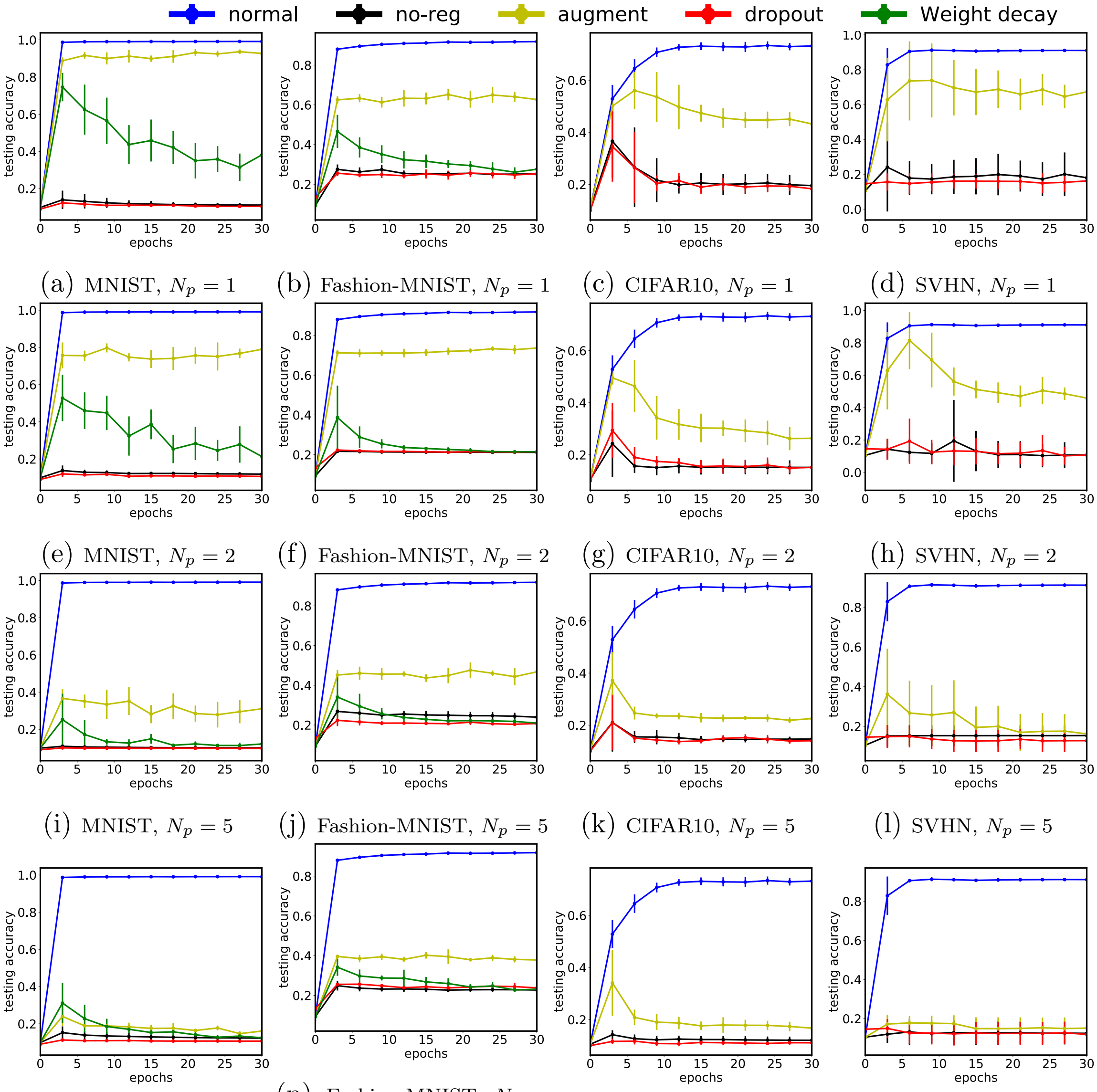
(a) $N_p = 1$

(b) $N_p = 2$

(c) $N_p = 5$

(d) $N_p = 10$

Figure A-4: Attacked training images for SVHN dataset.



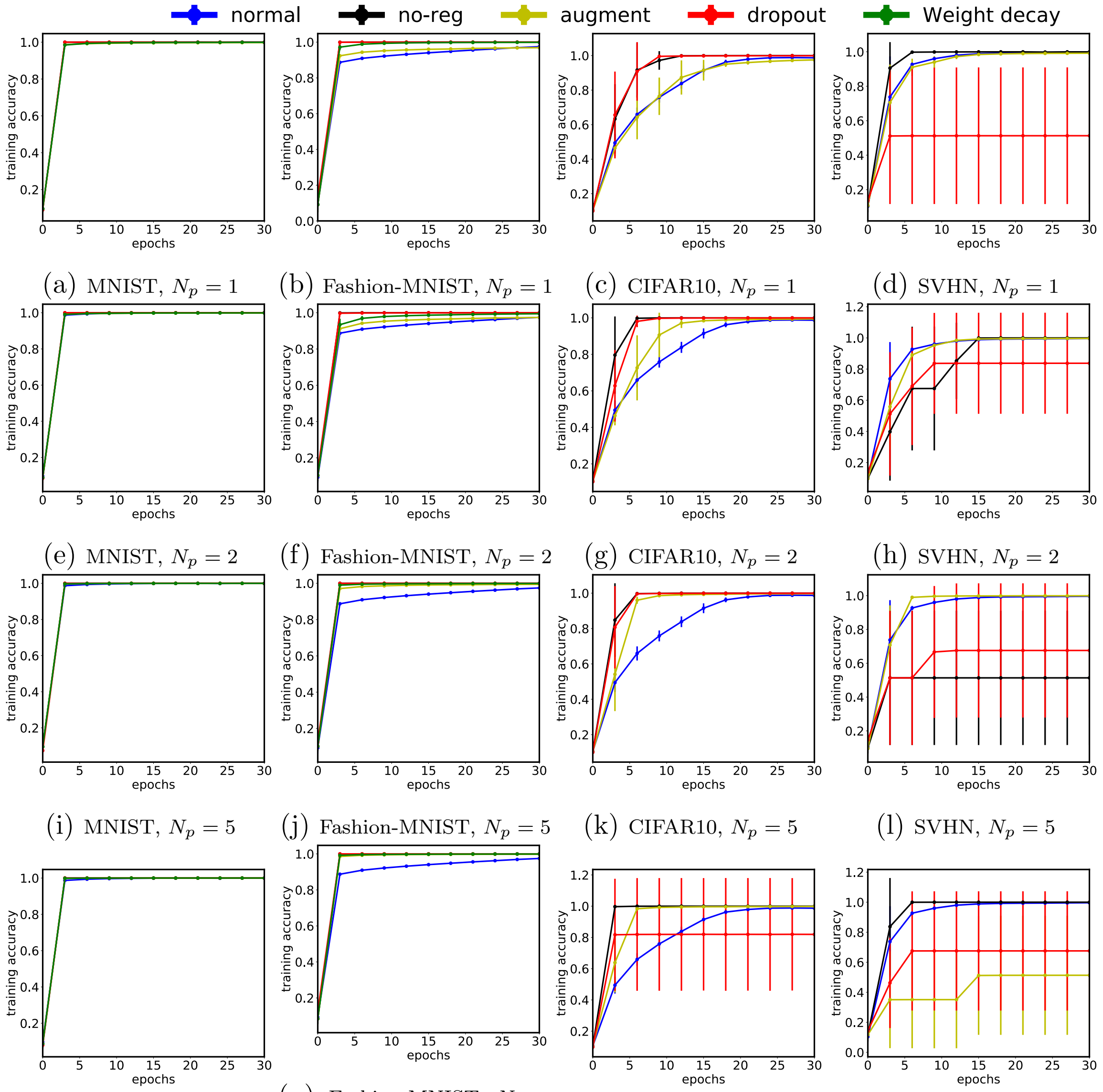
(a) MNIST, $N_p = 1$ (b) Fashion-MNIST, $N_p = 1$ (c) CIFAR10, $N_p = 1$ (d) SVHN, $N_p = 1$

(e) MNIST, $N_p = 2$ (f) Fashion-MNIST, $N_p = 2$ (g) CIFAR10, $N_p = 2$ (h) SVHN, $N_p = 2$

(i) MNIST, $N_p = 5$ (j) Fashion-MNIST, $N_p = 5$ (k) CIFAR10, $N_p = 5$ (l) SVHN, $N_p = 5$

(m) MNIST, $N_p = 10$ (n) Fashion-MNIST, $N_p = 10$ (o) CIFAR10, $N_p = 10$ (p) SVHN, $N_p = 10$

Figure A-5: Testing accuracy with increasing training epochs for different regularization methods for $N_p = 1, 2, 5, 10$ attack. *normal* refer to no attack scenario in training data and *no-reg* refers to the case where no explicit regularization was used. Experiments were repeated 5 times.



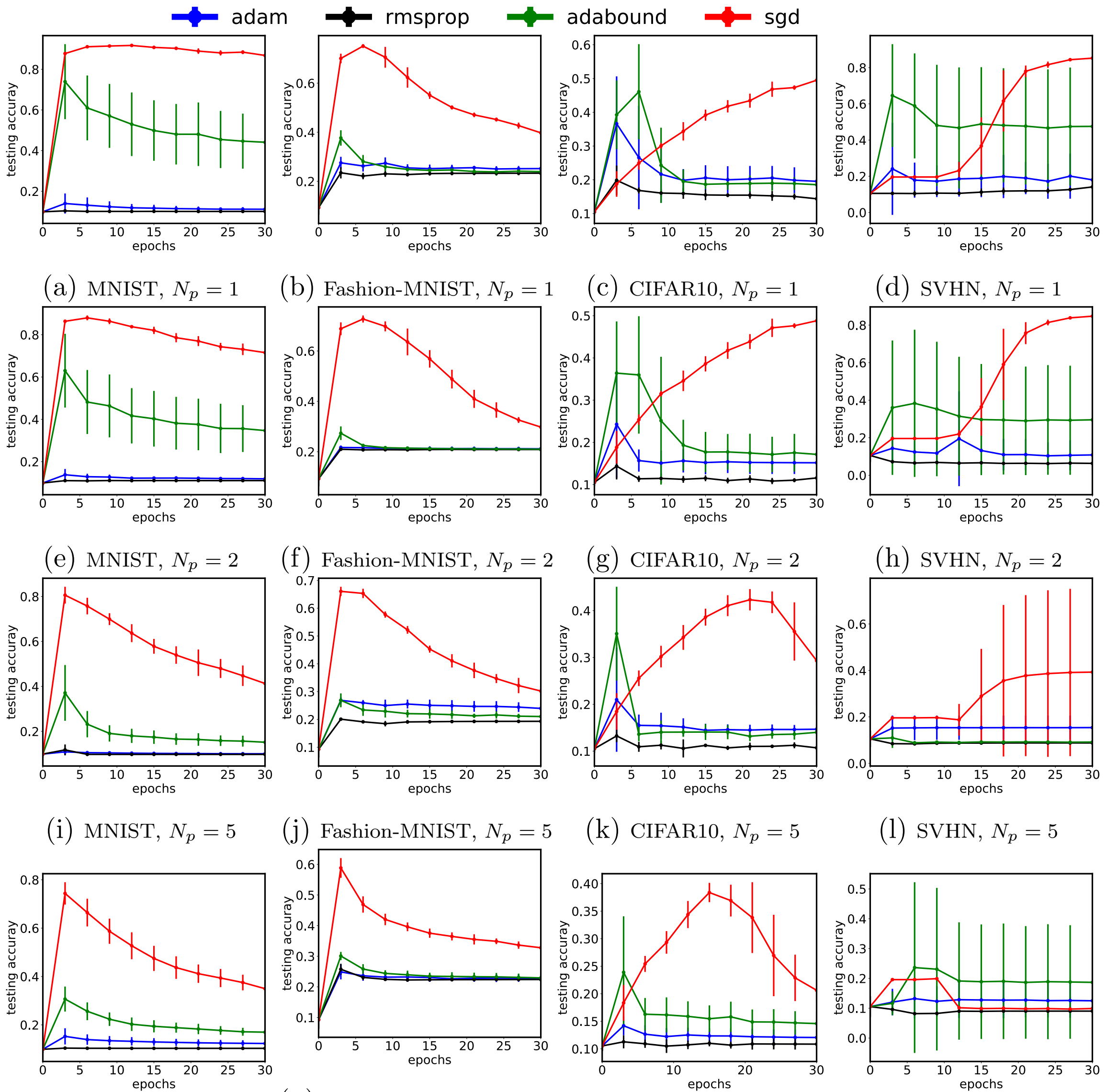
(a) MNIST, $N_p = 1$ (b) Fashion-MNIST, $N_p = 1$ (c) CIFAR10, $N_p = 1$ (d) SVHN, $N_p = 1$

(e) MNIST, $N_p = 2$ (f) Fashion-MNIST, $N_p = 2$ (g) CIFAR10, $N_p = 2$ (h) SVHN, $N_p = 2$

(i) MNIST, $N_p = 5$ (j) Fashion-MNIST, $N_p = 5$ (k) CIFAR10, $N_p = 5$ (l) SVHN, $N_p = 5$

(m) MNIST, $N_p = 10$ (n) Fashion-MNIST, $N_p = 10$ (o) CIFAR10, $N_p = 10$ (p) SVHN, $N_p = 10$

Figure A-6: Corresponding training accuracy for different regularization methods show almost 100% performance indicating overfitting. Experiments were repeated 5 times.



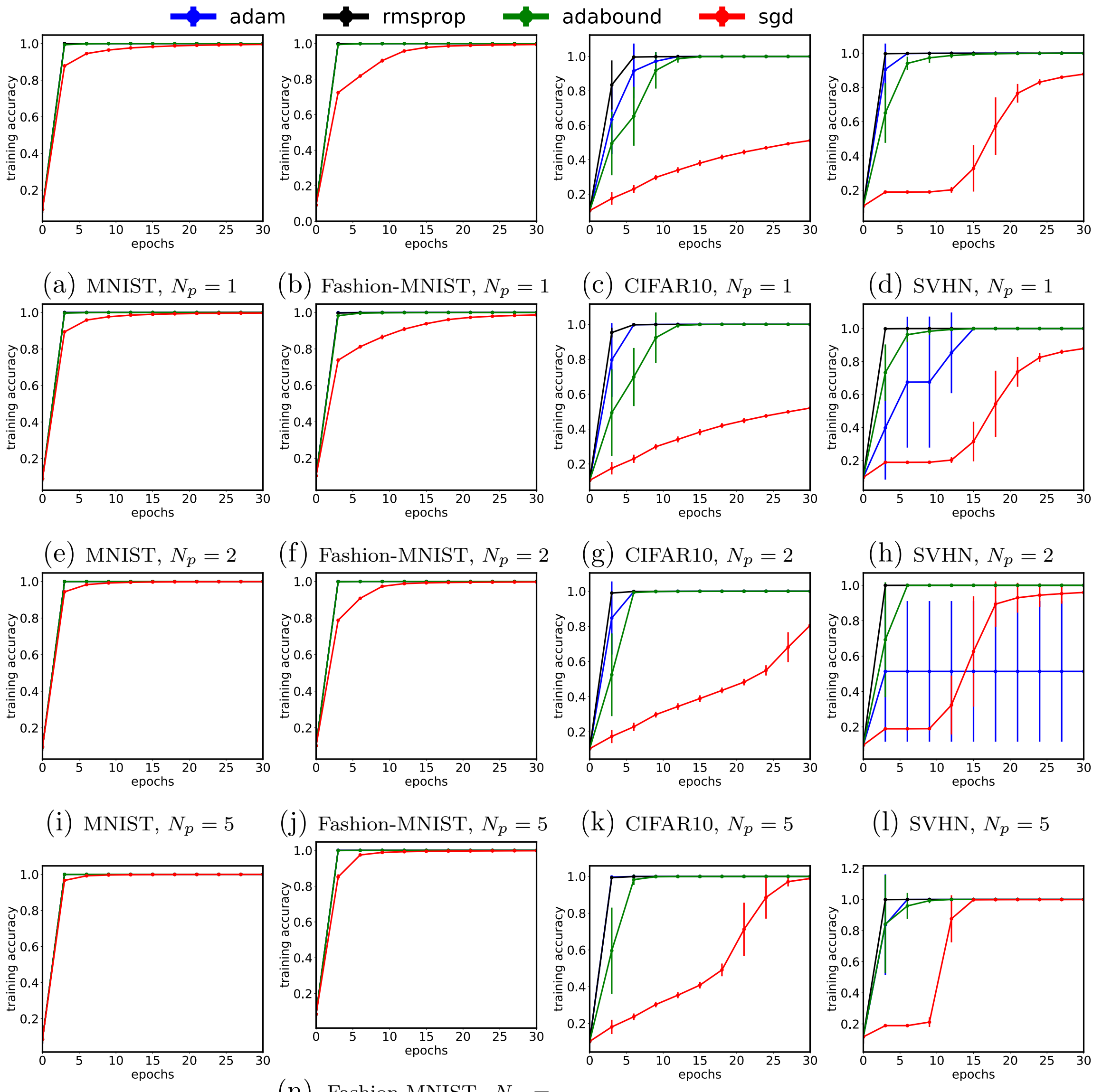
(a) MNIST, $N_p = 1$ (b) Fashion-MNIST, $N_p = 1$ (c) CIFAR10, $N_p = 1$ (d) SVHN, $N_p = 1$

(e) MNIST, $N_p = 2$ (f) Fashion-MNIST, $N_p = 2$ (g) CIFAR10, $N_p = 2$ (h) SVHN, $N_p = 2$

(i) MNIST, $N_p = 5$ (j) Fashion-MNIST, $N_p = 5$ (k) CIFAR10, $N_p = 5$ (l) SVHN, $N_p = 5$

(m) MNIST, $N_p = 10$ (n) Fashion-MNIST, $N_p = 10$ (o) CIFAR10, $N_p = 10$ (p) SVHN, $N_p = 10$

Figure A-7: Testing accuracy with increasing training epochs for different optimization methods under $N_p = 1, 2, 5, 10$ attack. SGD shows better robustness than other adaptive methods. Experiments were repeated 5 times.



(a) MNIST, $N_p = 1$ (b) Fashion-MNIST, $N_p = 1$ (c) CIFAR10, $N_p = 1$ (d) SVHN, $N_p = 1$

(e) MNIST, $N_p = 2$ (f) Fashion-MNIST, $N_p = 2$ (g) CIFAR10, $N_p = 2$ (h) SVHN, $N_p = 2$

(i) MNIST, $N_p = 5$ (j) Fashion-MNIST, $N_p = 5$ (k) CIFAR10, $N_p = 5$ (l) SVHN, $N_p = 5$

(m) MNIST, $N_p = 10$ (n) Fashion-MNIST, $N_p = 10$ (o) CIFAR10, $N_p = 10$ (p) SVHN, $N_p = 10$

Figure A-8: Corresponding training accuracy for different optimization methods under our proposed attack show almost 100% accuracy.

Appendix B

Publications

B.1 Publications Related to This Thesis

B.1.1 International Conference

1. **Subhajit Chaudhury** and Toshihiko Yamasaki, ”Investigating Generalization in Neural Networks Under Optimally Evolved Training Perturbations,” ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 2020, pp. 3617-3612, doi: 10.1109/ICASSP40776.2020.9053263.

2. **Subhajit Chaudhury**, “Understanding Generalization in Neural Networks for Robustness against Adversarial Vulnerabilities.” Proceedings of the AAAI Conference on Artificial Intelligence, 34(10), 13714-13715. https://doi.org/10.1609/aaai.v34i10.7129 [**Won scholarship for AAAI to attend doctoral consortium**]

B.1.2 Domestic Conference

1. **Subhajit Chaudhury** and Toshihiko Yamasaki, “Adversarial Attack during Learning”, MIRU 2019. [**Won best student paper, honorable mention**]

B.2 Publications Not Related to This Thesis

B.2.1 International Journal

1. Hiya Roy, **Subhajit Chaudhury**, Toshihiko Yamasaki, Tatsuaki Hashimoto, "Toward Better Planetary Surface Exploration by Mars Orbital Imagery Inpainting", IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (JSTARS), 2020.

2. Sourav Mishra, **Subhajit Chaudhury**, Hideaki Imaizumi, Toshihiko Yamasaki, "Robustness of DL models in Dermatological evaluation: A critical assessment", IEICE Trans. on Information and Systems, 2020.

B.2.2 International Conference and Symposium

1. Sourav Mishra, **Subhajit Chaudhury**, Hideaki Imaizumi and Toshihiko Yamasaki, "Assessing Robustness of Deep learning Methods in Dermatological Workflow", ACM Conference on Health Inference and Learning (CHIL), 2020 [**Spotlight paper**].

2. Roy H., **Chaudhury S.**, Yamasaki T., DeLatte D.M., Ohtake M., Hashimoto T., Lunar surface image restoration using U-Net based deep neural networks, IEEE International Conference on Computational Photography 2019 (Poster)

3. Roy H., **Chaudhury S.**, Yamasaki T., DeLatte D.M., Ohtake M., Hashimoto T., Lunar surface image restoration using U-Net based deep neural networks, 50th Lunar and Planetary Science Conference 2019, `https://www.hou.usra.edu/meetings/lpsc2019/pdf/2656.pdf`

B.2.3 Book Chapter

1. Roy H., **Chaudhury S.**, Yamasaki T., Hashimoto T., Machine Learning for Planetary Science, 1st Edition, Chapter 10: Enhancing Spatial Resolution of Remotely Sensed Imagery Using Deep Learning and/or Data Restoration, to

be published by ”Elsevier Science and Technology Books” on 1st March 2021, https://www.elsevier.com/books/machine-learning-for-planetary-science/helbert/978-0-12-818721-0

Bibliography

[1] *PyTurboJPEG*, 1.4.1 edition.

[2] Jakob Abeßer. A review of deep learning based methods for acoustic scene classification. *Applied Sciences*, 10(6), 2020.

[3] Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, and Mario Marchand. Domain-adversarial neural networks. *arXiv preprint arXiv:1412.4446*, 2014.

[4] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

[5] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[6] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*, 2018.

[7] David Barber and Felix V Agakov. The im algorithm: a variational approach to information maximization. In *Advances in neural information processing systems*, page None, 2003.

[8] Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

[9] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and R Devon Hjelm. Mine: mutual information neural estimation. *arXiv preprint arXiv:1801.04062*, 2018.

[10] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010.

[11] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In *Advances in neural information processing systems*, pages 137–144, 2007.

[12] Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. *arXiv preprint arXiv:1206.6389*, 2012.

[13] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

[14] Léon Bottou and Olivier Bousquet. The tradeoffs of large scale learning. In *Advances in neural information processing systems*, pages 161–168, 2008.

[15] Jacob Buckman, Aurko Roy, Colin Raffel, and Ian Goodfellow. Thermometer encoding: One hot way to resist adversarial examples. In *International Conference on Learning Representations*, 2018.

[16] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE, 2017.

[17] Rich Caruana, Steve Lawrence, and C Lee Giles. Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In *Advances in neural information processing systems*, pages 402–408, 2001.

[18] Subhajit Chaudhury and Toshihiko Yamasaki. Investigating generalization in neural networks under optimally evolved training perturbations. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3617–3612. IEEE, 2020.

[19] Subhajit Chaudhury and Toshihiko Yamasaki. Investigating generalization in neural networks under optimally evolved training perturbations. *International Conference on Acoustics, Speech, and Signal*, 2020.

[20] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[21] Pin-Yu Chen, Yash Sharma, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. Ead: elastic-net attacks to deep neural networks via adversarial examples. *arXiv preprint arXiv:1709.04114*, 2017.

[22] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[23] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[24] Gavin Weiguang Ding, Luyu Wang, and Xiaomeng Jin. AdverTorch v0.1: An adversarial robustness toolbox based on pytorch. *arXiv preprint arXiv:1902.07623*, 2019.

[25] Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1019–1028. JMLR. org, 2017.

[26] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193, 2018.

[27] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

[28] Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. Exploring the landscape of spatial robustness. In *International Conference on Machine Learning*, pages 1802–1811, 2019.

[29] Samuel G Finlayson, Hyung Won Chung, Isaac S Kohane, and Andrew L Beam. Adversarial attacks against medical deep learning systems. *arXiv preprint arXiv:1804.05296*, 2018.

[30] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

[31] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[32] Tom Goldstein, Christoph Studer, and Richard Baraniuk. A field guide to forward-backward splitting with a fasta implementation. *arXiv preprint arXiv:1411.3406*, 2014.

[33] Gene H Golub, Per Christian Hansen, and Dianne P O'Leary. Tikhonov regularization and total least squares. *SIAM Journal on Matrix Analysis and Applications*, 21(1):185–194, 1999.

[34] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[35] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[36] Ian J Goodfellow, Oriol Vinyals, and Andrew M Saxe. Qualitatively characterizing neural network optimization problems. *arXiv preprint arXiv:1412.6544*, 2014.

[37] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pages 5767–5777, 2017.

[38] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens Van Der Maaten. Countering adversarial images using input transformations. *arXiv preprint arXiv:1711.00117*, 2017.

[39] David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems*, pages 2450–2462, 2018.

[40] Nikolaus Hansen. The cma evolution strategy: A tutorial. *arXiv preprint arXiv:1604.00772*, 2016.

[41] Moritz Hardt, Benjamin Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. *arXiv preprint arXiv:1509.01240*, 2015.

[42] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[43] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

[44] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[45] Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural Computation*, 9(1):1–42, 1997.

[46] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[47] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[48] Jörn-Henrik Jacobsen, Jens Behrmann, Richard Zemel, and Matthias Bethge. Excessive invariance causes adversarial vulnerability. *arXiv preprint arXiv:1811.00401*, 2018.

[49] Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8018–8025, 2020.

[50] Daniel S Kermany, Michael Goldbaum, Wenjia Cai, Carolina CS Valentim, Huiying Liang, Sally L Baxter, Alex McKeown, Ge Yang, Xiaokang Wu, Fangbing Yan, et al. Identifying medical diagnoses and treatable diseases by image-based deep learning. *Cell*, 172(5):1122–1131, 2018.

[51] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

[52] Nitish Shirish Keskar and Richard Socher. Improving generalization performance by switching from adam to sgd. *arXiv preprint arXiv:1712.07628*, 2017.

[53] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[54] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[55] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1885–1894. JMLR. org, 2017.

[56] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

[57] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. *arXiv preprint arXiv:1512.09300*, 2015.

[58] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[59] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In *Advances in Neural Information Processing Systems*, pages 6389–6399, 2018.

[60] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In *Neural Information Processing Systems*, 2018.

[61] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1925–1934, 2017.

[62] Liangchen Luo, Yuanhao Xiong, Yan Liu, and Xu Sun. Adaptive gradient methods with dynamic bound of learning rate. *arXiv preprint arXiv:1902.09843*, 2019.

[63] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[64] Dongyu Meng and Hao Chen. Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 135–147. ACM, 2017.

[65] Sourav Mishra, Subhajit Chaudhury, Hideaki Imaizumi, and Toshihiko Yamasaki. Assessing robustness of deep learning methods in dermatological workflow. *arXiv preprint arXiv:2001.05878*, 2020.

[66] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1765–1773, 2017.

[67] Luis Muñoz-González, Battista Biggio, Ambra Demontis, Andrea Paudice, Vasin Wongrassamee, Emil C Lupu, and Fabio Roli. Towards poisoning of deep learning algorithms with back-gradient optimization. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 27–38. ACM, 2017.

[68] Nina Narodytska and Shiva Prasad Kasiviswanathan. Simple black-box adversarial perturbations for deep networks. *arXiv preprint arXiv:1612.06299*, 2016.

[69] Yurii Nesterov. A method for unconstrained convex minimization problem with the rate of convergence o (1/kˆ 2). In *Doklady an ussr*, volume 269, pages 543–547, 1983.

[70] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. In *Advances in Neural Information Processing Systems*, pages 5947–5956, 2017.

[71] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv preprint arXiv:1412.6614*, 2014.

[72] Matthew Olson, Abraham Wyner, and Richard Berk. Modern neural networks generalize on small data sets. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 3619–3628. Curran Associates, Inc., 2018.

[73] OpenCV. *The OpenCV Reference Manual*, 2.4.13.7 edition, April 2014.

[74] Utku Ozbulak, Arnout Van Messem, and Wesley De Neve. Impact of adversarial examples on deep learning models for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 300–308. Springer, 2019.

[75] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pages 372–387. IEEE, 2016.

[76] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 582–597. IEEE, 2016.

[77] Magdalini Paschali, Sailesh Conjeti, Fernando Navarro, and Nassir Navab. Generalizability vs. robustness: adversarial examples for medical imaging. *arXiv preprint arXiv:1804.00504*, 2018.

[78] Guillermo Valle Pérez, Ard A Louis, and Chico Q Camargo. Deep learning generalizes because the parameter-function map is biased towards simple functions. *arXiv preprint arXiv:1805.08522*, 2018.

[79] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

[80] Aaditya Prakash, Nick Moran, Solomon Garber, Antonella DiLillo, and James Storer. Deflecting adversarial attacks with pixel deflection. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8571–8580, 2018.

[81] Ning Qian. On the momentum term in gradient descent learning algorithms. *Neural networks*, 12(1):145–151, 1999.

[82] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[83] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred A Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. *arXiv preprint arXiv:1806.08734*, 2018.

[84] Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. *arXiv preprint arXiv:1904.09237*, 2019.

[85] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[86] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[87] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[88] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[89] Sara Sabour, Yanshuai Cao, Fartash Faghri, and David J Fleet. Adversarial manipulation of deep representations. *arXiv preprint arXiv:1511.05122*, 2015.

[90] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 618–626, 2017.

[91] Ali Shafahi, W Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. Poison frogs! targeted clean-label poisoning attacks on neural networks. In *Advances in Neural Information Processing Systems*, pages 6103–6113, 2018.

[92] Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

[93] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[94] Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. *arXiv preprint arXiv:1710.10766*, 2017.

[95] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[96] Jacob Steinhardt, Pang Wei W Koh, and Percy S Liang. Certified defenses for data poisoning attacks. In *Advances in neural information processing systems*, pages 3517–3529, 2017.

[97] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 2019.

[98] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[99] Thomas Tanay, Jerone TA Andrews, and Lewis D Griffin. Built-in vulnerabilities to imperceptible adversarial perturbations. *arXiv preprint arXiv:1806.07409*, 2018.

[100] T. Tieleman and G. Hinton. Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 2012.

[101] T Tieleman and G Hinton. Divide the gradient by a running average of its recent magnitude. coursera: Neural networks for machine learning. *Technical Report.*, 2017.

[102] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*, pages 1–5. IEEE, 2015.

[103] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

[104] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[105] Ashia C Wilson, Rebecca Roelofs, Mitchell Stern, Nati Srebro, and Benjamin Recht. The marginal value of adaptive gradient methods in machine learning. In *Advances in Neural Information Processing Systems*, pages 4148–4158, 2017.

[106] Lei Wu, Zhanxing Zhu, et al. Towards understanding generalization of deep learning: Perspective of loss landscapes. *arXiv preprint arXiv:1706.10239*, 2017.

[107] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. *arXiv preprint arXiv:1711.01991*, 2017.

[108] Kai Xu, Minghai Qin, Fei Sun, Yuhao Wang, Yen-Kuang Chen, and Fengbo Ren. Learning in the frequency domain. *2020 Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1737–1746, 2020.

[109] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*, 2017.

[110] Zhi-Qin John Xu, Yaoyu Zhang, and Yanyang Xiao. Training behavior of deep neural network in frequency domain. In *International Conference on Neural Information Processing*, pages 264–274. Springer, 2019.

[111] Dong Yin, Raphael Gontijo Lopes, Jon Shlens, Ekin Dogus Cubuk, and Justin Gilmer. A fourier perspective on model robustness in computer vision. In *Advances in Neural Information Processing Systems*, pages 13255–13265, 2019.

[112] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[113] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

[114] Chiyuan Zhang, Qianli Liao, Alexander Rakhlin, Karthik Sridharan, Brando Miranda, Noah Golowich, and Tomaso Poggio. Theory of deep learning iii: Generalization properties of sgd. *Center for Brains, Minds and Machines (CBMM), Tech. Rep*, 2017.

[115] Richard Zhang. Making convolutional networks shift-invariant again. In *ICML*, 2019.