UNIVERSITY OF CALIFORNIA,
IRVINE

Study of Stochastic and Sparse Neural Network Models with Applications

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Mathematics

by

Thu Dinh

Dissertation Committee:
Professor Jack Xin, Chair
Professor Hongkai Zhao
Professor Roman Vershynin

2020

DEDICATION

To Trang and my family.
To my future self: In 15 and 30 years, where would I be?

TABLE OF CONTENTS

Page

LIST OF FIGURES **v**

LIST OF TABLES **vi**

LIST OF ALGORITHMS **vii**

ACKNOWLEDGMENTS **viii**

CURRICULUM VITAE **ix**

ABSTRACT OF THE DISSERTATION **xi**

1 Introduction **1**
1.1 Residual Diffusion and the Senile Reinforced Random Walk Model 1
1.2 Regularization for Deep Neural Network Pruning 3

2 Enhanced Diffusivity in Perturbed Senile Reinforced Random Walk Models **11**
2.1 Nearest Neighbor SeRW Model 11
2.2 Perturbed SeRW Models 14
2.2.1 Deterministic Perturbation (Model I) 14
2.2.2 Stochastic Perturbation 14
2.3 Main Results 15
2.4 Proofs of Main Results 18
2.4.1 Theorem 2.3: Existence of Positive Diffusion Constant 18
2.4.2 Rate of Convergence 20
2.4.3 Theorem 2.4: Existence of Positive Diffusion Constant 21
2.4.4 Random Variables with Infinite Expectation 24
2.4.5 Proof of Theorem 2.5 28
2.4.6 Theorem 6: Results in Higher Dimensions 29

3 The Relaxed Variable Splitting Method **33**
3.1 Preliminaries 33
3.1.1 The One-layer Non-overlap Network 33
3.1.2 Modification with Binarized Activation 34
3.1.3 The Relaxed Variable Splitting Method 36

3.1.4 Comparison with ADMM . . . 38
3.2 Convergence Results . . . 39
3.2.1 The One-layer Model with ReLU Activation . . . 39
3.2.2 The One-layer Model with Binarized ReLU Activation . . . 41
3.3 Proof of the First Convergence Result . . . 43
3.3.1 Proof of Lemma 3 . . . 45
3.3.2 Proof of Lemma 6 . . . 47
3.3.3 Proof of Lemma 7 . . . 48
3.3.4 Proof of Theorem 3.1 . . . 49
3.4 Proof of the Second Convergence Result . . . 52
3.4.1 Proof of Lemma 10 . . . 54
3.4.2 Proof of Lemma 11 . . . 57
3.4.3 Proof of Lemma 12 . . . 58
3.4.4 Proof of Lemma 13 . . . 59
3.4.5 Proof of Theorem 3.2 . . . 61
3.4.6 Proof of Corollary . . . 61
3.5 Numerical Experiments . . . 62

4 Generalization of RVSM 64
4.1 Structured Pruning and Multi-layer Network . . . 64
4.1.1 The Relaxed Group-wise Splitting Method . . . 64
4.1.2 Convergence Analysis . . . 66
4.1.3 Numerical Experiments . . . 67
4.1.4 On The Modern Approach to Pruning . . . 68
4.2 Pruning Robustly Trained Network . . . 70
4.2.1 Overview . . . 70
4.2.2 Regularity and Sparsity of the Feynman-Kac Formalism Principled Robust DNNs' Weights . . . 73
4.2.3 Numerical Results . . . 74
4.2.4 Beyond CIFAR10 . . . 79

5 Conclusion 81

Bibliography 83

Appendix A Appendix Title 90

LIST OF FIGURES

Page

1.1 The architecture of a no-overlap neural network 6
1.2 Histograms of the ResNet20's weights. 9

3.1 Geometry of the update of $\boldsymbol{w}^t$ and the corresponding $\boldsymbol{w}^{t+1}, \boldsymbol{v}^{t+1}$. 45
3.2 Worst case of the update on $\boldsymbol{w}^t$ 50
3.3 Geometry of $\boldsymbol{w}^t$ and $\boldsymbol{w}^*$ when $\|\boldsymbol{w}^t\| = \|\boldsymbol{w}^*\| = 1$. 54

4.1 Weights visualization 73
4.2 Histogram of weights. 74
4.3 Sparsity of En$_2$ResNet20 and ResNet38 under different parameters λ_1. 77
4.4 Accuracy of En$_2$ResNet20 and ResNet38 under different parameters λ_1. . . . 78
4.5 Channel norms of the AT ResNet20 under RGSM and ADMM. 79

LIST OF TABLES

Page

1.1 Size and number of parameters of some modern DNN's. 5

3.1 Sparsity and accuracy of RVSM under different penalties on VGG-16 on CIFAR10. 63

3.2 Comparison between ADMM and RVSM (ℓ_0) for ResNet18 training on the CIFAR10 dataset. 63

4.1 Accuracy and channel sparsity of RGSM on LeNet and MNIST, with parameters $\beta = 1$ and $\lambda = 1$. 68

4.2 Accuracy and channel sparsity of RGSM on ResNet18 and VGG16, on CIFAR10. 68

4.3 Performance of RGSM against some state-of-the-art pruning algorithms, for VGG-16 on CIFAR10. 70

4.4 Accuracy and sparsity of ResNet20 and En$_2$ResNet20 under RVSM, with $\lambda = 1.e - 6$ and varying β. 76

4.5 Accuracy and sparsity of different EnResNet20 under RGSM. 76

4.6 Accuracy and sparsity of En$_2$ResNet20 and ResNet38 under RVSM. 77

4.7 Contrasting ADMM versus RVSM (unstructured) and RGSM (channel) for the robustly trained ResNet20. 78

4.8 Accuracy and sparsity of different Ensembles of ResNet20's on the CIFAR100. 80

LIST OF ALGORITHMS

Page

1 RVSM 37

2 RVSCGD 37

3 RGSM 65

ACKNOWLEDGMENTS

I would like to express my gratitude to my advisor, professor Jack Xin. He gave me the inspiration to stay in the program when I already had other plans. It was his meaningful discussions and insights that guided me along the way and got me to where I am today.

I would like to thank professors Hongkai Zhao and Roman Vershynin for serving as my conmmittee members, and for their thoughtful inputs.

I would like to thank Donna McConnell, Dr. Michiel Kosters, and professor Patrick Guidotti for helping me through my toughest time in the department; professor Katya Krupchyk for her guidance in Real Analysis; and Aubrey Rudd for her help with all the registration and paperwork.

Without the help and guidance of the faculty at Cal Poly Pomona, I would not have made it into the program in the first place. And for that, I would like to thank professor Ioana Mihaila and Alan Krinik.

Finally, I would like to express my appreciation to my wife Trang, my sister-in-law Ann, and my family for their unconditional love and support.

This work was partially supported by NSF grants DMS-1522383, DMS-1854434, and IIS-1632935.

CURRICULUM VITAE

Thu Dinh

EDUCATION

Doctor of Philosophy in Mathematics **2020**
University of California, Irvine *Irvine, CA*

Master of Science in Mathematics **2018**
University of California, Irvine *Irvine, CA*

Bachelor and Master of Science in Mathematics **2015**
California Polytechnic State University, Pomona *Pomona, CA*

RESEARCH EXPERIENCE

AI Research Scientist **2020–**
Latent AI *Princeton, New Jersey*

AI Research Intern **2020–2020**
Latent AI *Princeton, New Jersey*

Graduate Research Assistant **2016–2020**
University of California, Irvine *Irvine, California*

TEACHING EXPERIENCE

Teaching Assistant **2016–2020**
University of California, Irvine *Irvine, California*

Graduate Teaching Assistant **2014–2015**
California Polytechnic State University, Pomona *Pomona, CA*

REFEREED JOURNAL PUBLICATIONS

Enhanced Diffusivity in Perturbed Senile Reinforced Random Walk Models **May 2020**
Asymptotic Analysis

Convergence of a Relaxed Variable Splitting Coarse Gradient Descent Method for Learning Sparse Weight Binarized Activation Neural Network **May 2020**
Frontiers in Applied Mathematics and Statistics

REFEREED CONFERENCE PUBLICATIONS

Convergence of a Relaxed Variable Splitting Method for Learning Sparse Neural Networks via ℓ_1, ℓ_0, and transformed-ℓ_1 Penalties **Sept 2020**
Intelligent Systems Conference 2020

AWARDS

Google Research Credit Grant **2019**
Google

Euler Award for Outstanding Promise as a Graduate Student **2019**
Department of Mathematics, University of California, Irvine

ABSTRACT OF THE DISSERTATION

Study of Stochastic and Sparse Neural Network Models with Applications

By

Thu Dinh

Doctor of Philosophy in Mathematics

University of California, Irvine, 2020

Professor Jack Xin, Chair

We study the diffusivity of random walks with transition probabilities depending on the number of consecutive traversals of the last traversed edge, the so called senile reinforced random walk (SeRW). In one dimension, the walk is known to be sub-diffusive with identity reinforcement function. We perturb the model by introducing a small probability δ of escaping the last traversed edge at each step. The perturbed SeRW model is diffusive for any $\delta > 0$, with enhanced diffusivity ($\gg O(\delta^2)$) in the small δ regime. We further study stochastically perturbed SeRW models by having the last edge escape probability of the form $\delta\,\xi_n$ with ξ_n's being independent random variables. Enhanced diffusivity in such models are logarithmically close to the so called residual diffusivity (positive in the zero δ limit), with diffusivity between $O\left(\frac{1}{|\log\delta|}\right)$ and $O\left(\frac{1}{\log|\log\delta|}\right)$. Finally, we generalize our results to higher dimensions where the unperturbed model is already diffusive. The enhanced diffusivity can be as much as $O(\log^{-2}\delta)$.

Regularization of deep neural networks (DNN's) is one of the effective complexity reduction methods to improve efficiency and generalizability. We consider the problem of regularizing a one hidden layer convolutional neural network with ReLU activation function via gradient descent under sparsity promoting penalties. It is known that when the input data is Gaussian distributed, no-overlap networks (without penalties) in regression problems with ground truth can be learned in polynomial time at high probability. We propose a Relaxed Variable Splitting Method (RVSM), integrating thresholding and gradient descent to overcome the non-smoothness in the associated loss function. The sparsity in network weight is realized during the optimization (training) process. We prove that under ℓ_1, ℓ_0, and transformed-ℓ_1 penalties, no-overlap networks can be learned with high probability, and the iterative weights converge to a global limit which is a transformation of the true weight under a novel thresholding operation. Numerical experiments confirm theoretical finding, and compare the accuracy and sparsity trade-off among the penalties. On the CIFAR10 dataset, RVSM can sparsify ResNet18 up to 93.70%, with less than 0.2% loss in accuracy.

Finally, we generalize the RVSM algorithm to structured pruning, with applications to adversarial training. With structure sparsity, a DNN can be effectively pruned off without sacrificing performance, resulting in both smaller model size and the number of floating point operations. Furthermore, DNN's security and compression are two crucial tasks for deploying secure A.I. applications in resource-limited environments, such as self-driving cars or facial recognition on mobile devices. Traditionally, sparsity and robustness have been addressed separately, and not many pruning methods are known to perform well on robustly trained DNN's. We modify and integrate RVSM into the adversarial training process, and show that one can create a model that is both robust and sparse. On the CIFAR10 dataset, one can ensemble a model similar in size to ResNet38, but with over 40% channel sparisty (thus can be reduced in size accordingly), and better performance in both natural accuracy and accuracy against many standard adversarial attacks.

Chapter 1

Introduction

1.1 Residual Diffusion and the Senile Reinforced Random Walk Model

Enhanced diffusivity arises in large scale fluid transport through chaotic and turbulent flows, and has been studied for nearly a century, see [60, 31, 30, 4, 18, 46, 51, 43] among others. It refers to the much larger macroscopic effective diffusivity (D^E) than the microscopic molecular diffusivity (D_0) as the latter approaches zero. An example of smooth chaotic flow is the time periodic Hamiltonian flow ($X = (x, y) \in \mathbb{R}^2$):

$$\boldsymbol{v}(X,t) = (\cos(y), \cos(x)) + \theta\ \cos(t)\ (\sin(y), \sin(x)), \quad \theta \in (0, 1]. \tag{1.1}$$

The first term of (1.1) is a steady flow consisting of periodic arrays of counter-rotating vortices, and the second term is a time periodic perturbation that injects an increasing amount of disorder into the flow trajectories as θ becomes larger. At $\theta = 1$, the flow is fully mixing, and empirically sub-diffusive [80]. The flow (1.1) is one of the simplest models of chaotic advection in Rayleigh-Bénard experiment [8]. The motion of a diffusing particle in the flow (1.1) satisfies the stochastic differential equation (SDE):

$$dX_t = \boldsymbol{v}(X_t, t)\, dt + \sqrt{2\, D_0}\, dW_t, \quad X(0) = (x_0, y_0) \in \mathbb{R}^2, \tag{1.2}$$

where W_t is the standard 2-dimensional Wiener process. The mean square displacement in the unit direction e at large times is given by [3]:

$$\lim_{t \uparrow +\infty} E(|(X(t) - X(0)) \cdot e|^2)/t = D^E, \tag{1.3}$$

where $D^E = D^E(D_0, e, \theta) > D_0$ is the effective diffusivity. Numerical simulations [4, 51, 43] based on the associated Fokker-Planck equations (or cell problems of homogenization [3]) suggest that at $e = (1, 0)$, $\theta = 1$, $D^E = O(1)$ as $D_0 \downarrow 0$, the *residual diffusivity* emerges. In fact, $D^E = O(1)$ for $e = (0, 1)$ and a range of values in $\theta \in (0, 1)$ as well [43]. Recently, computation of (1.2)-(1.3) by structure preserving schemes [68] reveals residual diffusivity also for a time stochastic version of (1.1). At $\theta = 0$, enhanced D^E scales as $O(\sqrt{D_0}) \gg D_0$ as $D_0 \downarrow 0$, see [17, 26, 53] for various proofs and generalizations.

Motivated by enhanced diffusion in advecting fluids, we are interested in the enhanced diffusion phenomenon in discrete stochastic dynamics such as random walk models with some memory or tendency to return. The memory effects on a walker induce a slowdown of transport (movement) similar to spinning vortices in fluid flows. We shall add a small probability of symmetric random walk and examine the large time behavior of the second moment, in similar spirit to (1.3). The first work along this line of inquiry is [44] where the baseline model is the so called elephant random walk model with stops (ERWS) [56, 33]. The ERWS is non-Markovian and exhibits sub-diffusive, diffusive and super-diffusive regimes. The ERWS plays the role of flow (1.1). A transition from sub-diffusive to enhanced diffusive regime emerges with diffusivity strictly above that of the baseline model (hence residual diffusivity appears) as the added probability of symmetric random walk tends to zero [44].

In chapter 2, we study enhanced diffusivity by perturbing the Nearest-neighbor Senile Reinforced Random Walk model (SeRW, [28]). The model involves a standard random walk on $\mathbb{Z}^d$ and a reinforcement function $f : \mathbb{N} \to [-1, \infty)$. The walk $\{S_n\}_{n \geq 0}$ starts at the origin and initially steps to one of the $2\,d$ nearest neighbors with equal probability. Subsequent steps are defined by the number of times the current undirected edge has been traversed consecutively: If $\{S_{n-1}, S_n\}$ has been traversed m consecutive times in the immediate past, then the probability of traversing that edge in the next step is $\frac{1+f(m)}{2d+f(m)}$, with the rest of the possible $2d-1$ choices being equally likely. As soon as a new edge is traversed, the reinforcement ends on the previous edge and restarts on the new edge. For identity reinforcement function f, the walk is sub-diffusive in $d = 1$, and diffusive in higher dimension [28]. We analyze the asymptotics of the enhanced diffusivity when adding a variety of symmetric random walks at small probability. For multi-dimensional SeRW ($d \geq 2$), the enhancements come logarithmically close to residual diffusivity.

1.2 Regularization for Deep Neural Network Pruning

In the subsequent chapters, we study the theory of Deep neural networks (DNN) compression.

The theory of machine learning has been around since the 1950's (Turing, 1950 [62]), but it was not until the 2010's that the field really gained worldwide recognition. The discovery of DNN's has significantly changed the way of life in the last decade. Many highly technical tasks can now be done completely using DNN's such as speech recognition (Hinton et al., 2012 [27]), computer vision (Krizhevsky et al., 2016 [32]), and natural language processing (Dauphin et al., 2016 [15]).

In general, a neural network is a non-linear function that takes the input (image, sound, video,...) and outputs a "score" for the corresponding task. In most applications, the output can be a probability for classification tasks, a real number estimation for regression problems, or a coordinate prediction for object detection. For example: consider the digit recognition task, where a neural network takes in a picture of a handwritten digit, from 0 to 9, and gives a prediction for this number. A network for this classification problem can have L layers, each of which gives an output

$$\boldsymbol{x}^{i+1} = f^i(\boldsymbol{x}^i) := \sigma(\boldsymbol{W}^i\boldsymbol{x}^i + \boldsymbol{b}^i)$$

where $\boldsymbol{x}^i, \boldsymbol{W}^i$ and $\boldsymbol{b}^i$ are the input, weight (kernel), and bias of the i^{th} layer, for $1 \leq i \leq L$; and σ is the Rectified Linear Unit (ReLU) function, $\sigma(x) = \max\{x, 0\}$. Given an input $\boldsymbol{x}$ (a matrix of pixel values for an image), the output of the network is then

$$\boldsymbol{y} = f^L \circ ... \circ f^1(\boldsymbol{x})$$

In practice, f^L is usually a soft-max layer, and $\boldsymbol{y} \in \mathbb{R}^{10}$ is a vector of probability for the numbers 0 to 9. The component with highest value is then chosen to be the prediction of the network.

Training such networks is a problem of minimizing a high-dimensional non-convex and non-smooth objective function, and is often solved by simple first-order methods such as stochastic gradient descent. Nevertheless, the success of neural network training remains to be understood from a theoretical perspective. Progress has been made in simplified model problems. Shamir (2016) showed learning a simple one-layer fully connected neural network is hard for some specific input distributions [57]. Recently, several works ([61, 7]) focused on the geometric properties of loss functions, which is made possible by assuming that the input data distribution is Gaussian; and showed that stochastic gradient descent (SGD) with random or zero initialization is able to train a no-overlap neural network in polynomial time.

In the last decade, many different DNN designs were introduced. Taking advantage of the state-of-the-art computing power, there has been an upward trend in model size and number of parameters [35, 29, 24, 79, 58] (Table 1.1). Most notably, VGG16 is a network with over 500MB in size and 138 million trainable parameters.

Table 1.1: Size and number of parameters of some modern DNN's.

Model	Size	Parameters
LeNet	0.25MB	60K
MobileNet	16MB	4.2M
ResNet50	98MB	25.5M
NASNet	343MB	89M
VGG16	528MB	138M

An immediate issue is that DNN's are often over-parameterized with millions of parameters that contain lots of redundancies, which can cause over-fitting and poor generalization [74], besides spending unnecessary computational resources and storage space. On resource-limited devices such as mobile phones or tablets, it is important that one can reduce the model size and computational latency, while also maintaining reasonable performance. To achieve this goal, two common techniques are regularization and quantization.

In regularization, the network is (possibly) retrained and pruned in such a way that results in lots of zero components. A sparse network has a much smaller number of floating point operations (FLOPS) during computation, giving much faster inference rate. Networks can also be trained such that all the zero components occur in a certain structure (channel/filter/depth), which can then be effectively pruned off.

In quantization, all the weights and biases in the network are quantized to a certain (low) bit. In these reduced precision ranges, one can achieve faster inference rate by using optimized kernels like GEMMLOWP [20], Intel MKL-DNN [47], or TensorRT [54]. With 8-bit quantization, one can reduce the model size by a factor of 4 (from 32-bit floating point) without any noticeable loss to performance. Quantization is also critical if one wants to run a neural network on a hardware without floating point support, for example Digital Signal Processor chips.

Our research focuses on improving network sparsity and reducing latency via regularization. In this sub-area, many modern algorithms are based on an empirical technique called pruning [36, 23], where all the non-essential components can be zeroed out with minimal loss of performance [63, 49]. A typical workflow for network pruning involves three stages: (1) train the over-parameterized model (which can take up to days or weeks on some state-of-the-art networks); (2) prune the model based on some criterion; (3) fine-tuning the pruned model (which may involve some re-training). Recently, a surrogate ℓ_0 regularization approach based on a continuous relaxation of Bernoulli random variables in the distribution sense is introduced with encouraging results on small size image data sets [42]. This motivated our work here to study the deterministic regularization of ℓ_0 (and ℓ_1 penalty) via its Moreau envelope. The method we propose here combines all three stages of network pruning into one, and will be shown to have competitive performance with many state-of-the-art techniques.

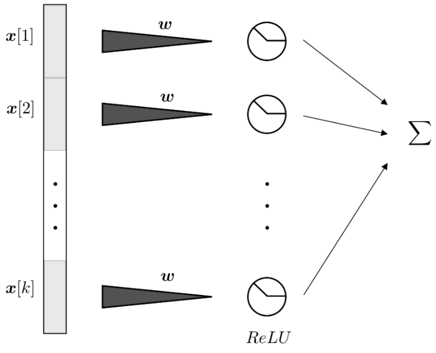


Figure 1.1: The architecture of a no-overlap neural network

In chapter 3, we propose a new method to sparsify DNN's called the Relaxed Variable Splitting Method (RVSM), and prove its convergence a simple one-layer network. The architecture of this network is illustrated in Figure 1.1, similar to [7]. We consider a convolution layer in which a sparse filter $\boldsymbol{w} \in \mathbb{R}^d$ is shared among k different hidden nodes. The input sample is $x \in \mathbb{R}^{kd}$. We assume the convolution filter is applied in a non-overlap way to k patches of the input: $\boldsymbol{x}[1], ..., \boldsymbol{x}[k]$, each with size d. We also assume that the input vectors $\boldsymbol{x}$ are i.i.d. Gaussian random vectors with zero mean and unit variance. The output of the network in Figure 1 is given by:

$$L(\boldsymbol{x};\boldsymbol{w}) = \frac{1}{k}\sum_{i=1}^{k}\sigma(\boldsymbol{w}\cdot\boldsymbol{x}[i]) \tag{1.4}$$

We address the realizable case, where the response training data is mapped from the input training data $\boldsymbol{x}$ by equation (1.4) with a ground truth unit weight vector $\boldsymbol{w}^*$. The input training data is generated by sampling n training points $\boldsymbol{x}^1, .., \boldsymbol{x}^n$ from a Gaussian distribution. The learning problem seeks $\boldsymbol{w}$ to solve the minimization problem:

$$\min_{\boldsymbol{w}} \frac{1}{n}\sum_{j=1}^{n}(L(\boldsymbol{x};\boldsymbol{w}) - L(\boldsymbol{x};\boldsymbol{w}^*))^2 \tag{1.5}$$

In the limit $n \to \infty$, this is equivalent to minimizing the population risk:

$$f(\boldsymbol{w}) := \mathbb{E}_{\boldsymbol{x}\sim\mathcal{G}}\left[(L(\boldsymbol{x};\boldsymbol{w}) - L(\boldsymbol{x};\boldsymbol{w}^*))^2\right] \tag{1.6}$$

As the training size is often large, we believe the problem of population loss minimization can sufficiently capture the key features of the network. We note that the iterative thresholding algorithms (IT) are commonly used for retrieving sparse signals ([14, 9, 6, 5, 76] and references therein). In high dimensional setting, IT algorithms provide simplicity and low computational cost, while also promote sparsity of the target vector. We shall investigate the convergence of training the network with simultaneous thresholding for the following objective function

$$\phi(\boldsymbol{w}) = f(\boldsymbol{w}) + \lambda\, P(\boldsymbol{w}) \tag{1.7}$$

where $f(\boldsymbol{w})$ is defined in (1.6), and P is ℓ_0, ℓ_1, or the transformed-ℓ_1 (Tℓ_1) function: a one parameter family of bilinear transformations composed with the absolute value function [52, 77]. When acting on vectors, the Tℓ_1 penalty interpolates ℓ_0 and ℓ_1 with thresholding in closed analytical form for any parameter value [76]. The ℓ_1 thresholding function is known as soft-thresholding [14, 16], and that of ℓ_0 the hard-thresholding [6, 5]. As pointed out in [42], it is beneficial to attain sparsity during the optimization (training) process.

We propose a Relaxed Variable Splitting Method (RVSM), which combines thresholding and gradient descent for minimizing the following augmented objective function

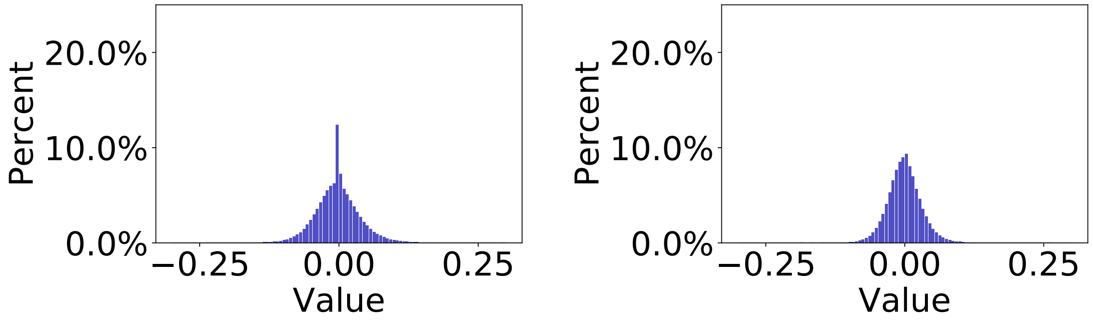
$$\boldsymbol{\mathcal{L}}_{\beta}(\boldsymbol{u}, \boldsymbol{w}) = f(\boldsymbol{w}) + \lambda\, P(\boldsymbol{u}) + \frac{\beta}{2}\, \|\boldsymbol{w} - \boldsymbol{u}\|^2$$

for a positive parameter β. We note in passing that minimizing $\boldsymbol{\mathcal{L}}_{\beta}$ in $\boldsymbol{u}$ recovers the original objective (1.7) with penalty P replaced by its Moreau envelope [50]. We shall prove that our algorithm (RVSM), which alternately minimizes $\boldsymbol{u}$ and $\boldsymbol{w}$, converges for ℓ_0, ℓ_1, and Tℓ_1 penalties to a global limit $(\bar{\boldsymbol{w}}, \bar{\boldsymbol{u}})$ with high probability. The limit $\bar{\boldsymbol{w}}$ is a novel shrinkage of the true weight $\boldsymbol{w}^*$ up to a scalar multiple, and the limit $\bar{\boldsymbol{u}}$ is a sparse approximation of $\boldsymbol{w}^*$.

In chapter 4, we extend RVSM to address two important topics in modern network compression: structured pruning and robust network compression.

While sparsifying a DNN can greatly reduce the number of floating point operations (FLOPs), in practice, it is also important to study the problem of structured pruning. If a Conv2D layer has a channel (or filter) that contains only zero's, that channel (or filter) can be safely pruned off. The resulting is both smaller in size and faster in inference rate, while still maintaining similar performance. This is especially important on resource-limited devices such as mobile phones or tablets. Many techniques were introduced to address channel sparsity, ranging from group-lasso SSL regularization [69], to channel-wise scaling factor training [40]. We will show that, with some modification, RVSM can be generalized to address many variations of structured pruning, with competitive performance against other state-of-the-art techniques.

Finally, we discuss the application of our pruning algorithm in adversarially trained network. An adversarial attack is a carefully crafted input that can fool a network's prediction, leaving the network vulnerable to malicious attackers. As a result, robust DNN's compression is a fundamental problem for secure AI applications in resource-constrained environments such as biometric verification, facial login on mobile devices, and computer vision tasks for the internet of things (IoT) [12, 71, 48]. Though compression and robustness have been separately addressed in recent years, it is important to explicitly have a method that can address both issues simultaneously, as their critical roles in the modern machine learning framework cannot be understated.



(a) NT

(b) AT

Figure 1.2: Histograms of the ResNet20's weights.

Adversarial training (AT) can create models that are more robust than natural training (NT) DNN's to attacks [45, 2]. However, AT models contain weights that are much less sparse than that of NT models. As shown in Fig. 1.2, start from the same default initialization in PyTorch, the NT ResNet20's weights are much sparser than that of the AT counterpart, for instance, the percent of weights that have magnitude less than 10^{-3} for NT and AT ResNet20 are 8.66% and 3.64% (averaged over 10 trials), respectively. Recently, a Feynman-Kac formalism principled ResNet ensemble was proposed in [65]; this is an AT algorithm that ensembles many identical ResNets and train in such a way that the resulting model with more small weights and more robust than a larger ResNet of similar size. We will incorporate RVSM into the AT process and show that such one can achieve a very sparse model that is also robust to standard adversarial attacks.

Chapter 2

Enhanced Diffusivity in Perturbed Senile Reinforced Random Walk Models

2.1 Nearest Neighbor SeRW Model

A *nearest-neighbor senile reinforced random walk* in $\mathbb{Z}^d$ is a sequence $\{S_n\}_{n\geq 0}$ of $\mathbb{Z}^d$-valued random variables on a probability space $(\Omega, \mathcal{F}, \mathbb{P}_f)$, with corresponding filtration $\{\mathcal{F}_n = \sigma(S_0, ..., S_n)\}_{n\geq 0}$, defined by:

- The walk begins at the origin of $\mathbb{Z}^d$, i.e. $S_0 = 0, \mathbb{P}_f$-almost surely.
- $\mathbb{P}_f(S_1 = x) = D(x)$, where $D(x) = (2d)^{-1}\mathbb{1}_{|x|=1}$
- For $n \in \mathbb{N}, e_n = \{S_{n-1}, S_n\}$ is an $\mathcal{F}_n$-measurable *undirected* edge and

$$m_n = \max\{k \geq 1 : e_{n-l+1} = e_n \text{ for all } 1 \leq l \leq k\}$$

is an $\mathbb{N}$-valued, $\mathcal{F}_n$-measurable random variable.

- For $n \in \mathbb{N}$ and $x \in \mathbb{Z}^d$ such that $|x| = 1$:

$$\mathbb{P}_f(S_{n+1} = S_n + x | \mathcal{F}_n) = \begin{cases} \frac{1+f(m_n)}{2d+f(m_n)}, & \text{if } \{S_n, S_n + x\} = e_n, \\ \frac{1}{2d+f(m_n)}, & \text{if } \{S_n, S_n + x\} \neq e_n, \end{cases}$$

We shall consider the case $f(m_n) = m_n$, and suppress the f dependence in the probability $\mathbb{P}_f$ notation. We shall refer to the analysis of SeRW model by Holmes and Sakai [28] and their main results without proofs.

Let $\tau = \sup\{n \geq 1 : S_m = 0 \text{ or } S_1 \; \forall m \leq n\}$ denote the number of times that the walk traverses the first edge before leaving that edge for the first time. Note that τ is not a stopping time (however $\tau + 1 = \inf\{n \geq 2 : S_n \neq S_{n-2}\}$ is a stopping time). Let N_x denote the number of times the walk S_n visits x. If $\mathbb{P}(N_x = \infty) = 1$ for all x, we say the walk is recurrent (I). If $\mathbb{P}(N_x = \infty) = 0$ for all x, we say the walk is transient (I). If $\mathbb{E}[N_x] = \infty$ for every x, we say the walk is recurrent (II), and if $\mathbb{E}[N_x] < \infty$ for all x, we say the walk is transient (II). Note that for the standard random walk, the two characterizations of recurrence/transience are equivalent, and the walk is recurrent in $d \leq 2$, and transient otherwise. For the senile reinforced random walks, the two notions need not be the same.

Theorem 2.1. *(Holmes and Sakai [28]) For f satisfying $\mathbb{P}_f(\tau = \infty) = 0$, but excluding the degenerate case where $d = 1$ and $f(1) = -1$, we have:*
(1) SeRW$_f$ is recurrent (I)/transient (I) if and only if SeRW$_0$ is recurrent (I)/transient (I).
(2) When $\mathbb{E}_f[\tau] < \infty$, SeRW$_f$ is recurrent (II)/transient (II) if and only if SeRW$_0$ is recurrent (II)/transient (II).
(3) When $\mathbb{E}_f[\tau] = \infty$, SeRW$_f$ is recurrent (II).

A consequence of this proposition is the following corollary:

Corollary 1. *The nearest-neighbor senile reinforced random walk with linear reinforcement of the form $f(m) = C\,m$ is recurrent (I), (II) when $d = 1, 2$ and transient (I) when $d > 2$. It is transient (II) for $d > 2$ if and only if $C < 2d - 1$.*

The diffusion constant is defined as $\nu = \lim_{n\to\infty} \mathbb{E}[|S_n|^2]$ (=1 for the standard random walk) whenever this limit exists. An important result of [28] is:

Theorem 2.2. *(Holmes and Sakai [28]) Suppose that there exists $\epsilon > 0$ and $\mathbb{E}[\tau^{1+\epsilon}] < \infty$. Then the walk is diffusive and the diffusion constant is given by*

$$\nu = \frac{\mathbb{P}(\tau \ \mathit{odd})}{1 - \frac{1}{d}\mathbb{P}(\tau \ \mathit{odd})} \frac{1}{\mathbb{E}[\tau]}. \tag{2.1}$$

The proof of Theorem 2.2 is based on the formula for the Green's function, and a Tauberian theorem, whose application requires the $(1 + \epsilon)$th moment of τ to be finite. Except for the degenerate case, it was shown in [28] that the result holds for all f by a time-change argument. When $\mathbb{E}[\tau] = \infty$, the right-hand side of (2.1) is zero, which suggest that the walk is sub-diffusive.

When $f(m) = m$, special hypergeometric functions are applicable and various well-known properties of these functions enable a proof of:

Proposition 2.2.1. *(Holmes and Sakai [28]) The diffusion constant ν of the nearest-neighbor senile random walk with reinforcement $f(l) = l$ satisfies $0 < \nu < 1$ when $d > 1$. For the one-dimensional nearest-neighbor model,*

$$\lim_{n\to\infty} \frac{\log n}{n} \mathbb{E}[|S_n|^2] = \frac{1 - \log 2}{2\log 2 - 1}.$$

Hence at $d = 1$, the walk is sub-diffusive, slower than diffusion by a logarithmic factor $(\log n)^{-1/2}$.

2.2 Perturbed SeRW Models

2.2.1 Deterministic Perturbation (Model I)

The one-dimensional model with $f(m) = m$ is sub-diffusive. This is partly due to the walk having a strong tendency to return to the last traversed edge. We add a small perturbation δ to the conditional probability of S_{n+1}:

$$\mathbb{P}(S_{n+1} = S_n + x|\mathcal{F}_n) = \begin{cases} \frac{1+m_n}{2+m_n} - \delta, & \text{if } \{S_n, S_n + x\} = e_n, \\ \frac{1}{2+m_n} + \delta, & \text{if } \{S_n, S_n + x\} \neq e_n. \end{cases}$$

In other words, at each step we add a small probability δ of escaping the last traversed edge, where $\delta > 0$ is deterministic. As $m_n \to \infty, \frac{1}{2+m_n} \to 0$. So if an edge has already been traversed consecutively too many times, the probability of escaping will be dominantly determined by δ. As a result, the perturbed model will gradually converge to a simplified model where the probability of returning to the last traversed edge is $1 - \delta$.

2.2.2 Stochastic Perturbation

Sequence of i.i.d. perturbations (Model II)

Let $(\xi_n)_{n\in\mathbb{N}}$ be a sequence of independent identically distributed (i.i.d.) non-negative random variables and we consider:

$$\mathbb{P}(S_{n+1} = S_n + x|\mathcal{F}_n) = \begin{cases} \max\{\frac{1+n}{2+n} - \delta\xi_n, 0\}, & \text{if } \{S_n, S_n + x\} = e_n, \\ \min\{\frac{1}{2+n} + \delta\xi_n, 1\}, & \text{if } \{S_n, S_n + x\} \neq e_n. \end{cases}$$

At each step, the random variable ξ_n takes a value, then the reinforcement is based on this value. We only assume that ξ_n is continuous with probability density function $f = f_{\xi_n}$.

Notice that if ξ_n takes any value greater than $\frac{1+n}{2+n}$, the walk will escape the last traversed edge on the $n+1^{th}$ turn. So in this model, the tail of the distribution function f provides a stronger chance of breaking out of the last traversed step, leading to more enhanced diffusion.

Sequence of independent perturbations (Model III)

To further enhance diffusivity, we shall consider the situation that $(\xi_n)_{n\in\mathbb{N}}$ are no longer i.i.d., but rather have n-dependent distributions.

$$\mathbb{P}(S_{n+1} = S_n + x|\mathcal{F}_n) = \begin{cases} \max\{\frac{1+n}{2+n} - \delta\xi_n, 0\}, & \text{if } \{S_n, S_n + x\} = e_n, \\ \min\{\frac{1}{2+n} + \delta\xi_n, 1\}, & \text{if } \{S_n, S_n + x\} \neq e_n, \end{cases}$$

For example, ξ_n's can have the same type of distribution and expectation, but with variance n^2. This modification will reinforce the probability of the walk breaking out of the last traversed edge. We only assume that $\mathbb{E}[\xi_n] < \infty$, for all n.

2.3 Main Results

The diffusivity from the perturbation (the simple symmetric random walk) similar to "molecular diffusivity" D_0 of (1.2) is $\nu_\delta = \delta^2$. We will show that, in all of our three models, the enhanced diffusivity is much greater than $O(\delta^2)$. Our main results are stated in the following theorems.

Theorem 2.3. *The deterministic perturbed model (I) is diffusive for any $\delta > 0$, and the diffusion constant is given by*

$$\nu = \frac{\mathbb{P}(\tau \ odd)}{\mathbb{P}(\tau \ even)\mathbb{E}[\tau]}. \tag{2.2}$$

Moreover,

$$\nu(\delta) = O\left(\frac{1}{|\log \delta|}\right) \quad as \quad \delta \to 0^+. \tag{2.3}$$

The formula (2.2) for ν is a direct result of Theorem 2.2. It is dramatic that the walk becomes diffusive for any value of $\delta > 0$. From proposition 2.2.1, the walk is sub-diffusive by an order of $\log n$. The added (deterministic) perturbation, no matter how small, is enough to create diffusivity.

To prove Theorem 2.3, we first verify that the model is diffusive by checking the condition of Theorem 2.2, then we find a lower bound for $\mathbb{E}[\tau]$ and show that the bound goes to ∞ as $n \to \infty$. A straightforward computation shows $1 \leq \frac{\mathbb{P}(\tau \text{ odd})}{\mathbb{P}(\tau \text{ even})} \leq 2$, which gives (2.2). This concludes the proof. Finally, we discuss the rate at which ν goes to zero as δ tends to zero.

Theorem 2.4. *The stochastic perturbed model (II) is diffusive for any $\delta > 0$, and the diffusion constant is also given by* (2.2). *Moreover,*
(i) If $\mathbb{E}[\xi_n] < \infty$, then $\nu(\delta) = O\left(\frac{1}{|\log \delta|}\right)$ as $\delta \to 0^+$.
(ii) If $\mathbb{E}[\xi_n] = \infty$, one can construct ξ_n so that $\nu(\delta) = O(\frac{1}{\log|\log \delta|})$ as $\delta \to 0^+$.

Similar to the deterministic case, the stochastic perturbed model is still not strong enough to sustain residual diffusivity. We can, however, reduce the rate at which ν converges to 0. If ξ_n has infinite expected value (fat tail), then ξ_n is more likely to attain very large values, and the walk is less likely to get stuck. The maximal enhancement on $\nu(\delta)$ is $O(\frac{1}{\log|\log \delta|})$.

Theorem 2.5. *The stochastic perturbed model (III) is diffusive for any $\delta > 0$. The diffusion constant is also given by (2.2) with $\nu(\delta) = O\left(\frac{1}{|\log \delta|}\right)$ as $\delta \to 0^+$.*

The proofs of the three theorems above are based on Theorem 2.2 to show diffusivity and the calculation of the diffusion constant ν. Our approach is elementary and relies heavily on the computation of the quantity $\mathbb{P}(\tau \geq n)$. The absence of residual diffusivity and the rate of convergence are obtained via asymptotic analysis in the small δ regime.

Theorem 2.6. *When the baseline diffusive SeRW model on $\mathbb{Z}^d$ $(d \geq 2)$ is perturbed into models (I, II, III), we have the following:*
(i) Under model I, the walk has a linearly enhanced diffusivity:

$$\nu_\delta = \nu_0 + O(\delta),$$

where ν_0 is the diffusivity of the unpeturbed model.
(ii) Under models II and III, if $\mathbb{E}[\xi_n] < \infty$, for all n, the walk has the same linear enhanced diffusivity as in model I.
(iii) Under models II and III, if $\mathbb{E}[\xi_n] = \infty$, for all n, one can construct ξ_n to achieve the following enhanced diffusivity rates:

$$\begin{aligned} &(a) \quad \nu_\delta = \nu_0 + O(\delta\,|\log \delta|), \\ &(b) \quad \nu_\delta = \nu_0 + O(\delta^j), \quad \textit{for some} \quad j \in (0,1), \\ &(c) \quad \nu_\delta = \nu_0 + O(\log^{-2} \delta). \end{aligned}$$

2.4 Proofs of Main Results

2.4.1 Theorem 2.3: Existence of Positive Diffusion Constant

First we verify the perturbed model is diffusive. Notice that

$$\mathbb{P}(\tau = 1) = \frac{1}{3} + \delta \qquad \text{and} \qquad \mathbb{P}(\tau = n) = \left[\prod_{k=2}^{n}\left(\frac{k}{k+1} - \delta\right)\right]\left(\frac{1}{n+2} + \delta\right)$$

for $n \geq 2$. We will show there exists $\epsilon > 0$ such that $\mathbb{E}[\tau^{1+\epsilon}] < \infty$ and apply Theorem 2.2. The following is an upper bound for $\mathbb{P}(\tau = n)$ when $n \geq 2$:

$$\begin{aligned}
\mathbb{P}(\tau = n) &= \left[\prod_{k=2}^{n}\left(\frac{k}{k+1} - \delta\right)\right]\left(\frac{1}{n+2} + \delta\right) \\
&= \left(\frac{2}{3} - \delta\right)\left(\frac{3}{4} - \delta\right)...\left(\frac{n}{n+1} - \delta\right)\left(\frac{1}{n+2} + \delta\right) \\
&= \frac{2(1 - \frac{3\delta}{2})3(1 - \frac{4\delta}{3})...n(1 - \frac{(n+1)\delta}{n})}{3 \cdot 4... \cdot (n+1)}\left(\frac{1}{n+2} + \delta\right) \\
&\leq \frac{2}{n+1}e^{-\frac{3\delta}{2}}e^{-\frac{4\delta}{3}}...e^{-\frac{(n+1)\delta}{n}}\left(\frac{1}{n+2} + \delta\right) \\
&= \frac{2}{n+1}\exp\left\{-\sum_{k=2}^{n}\delta\left(1 + \frac{1}{k}\right)\right\}\left(\frac{1}{n+2} + \delta\right) \\
&\leq \frac{2}{n+1}\exp\left\{\delta(-n + 1 - \log n + 1)\right\}\left(\frac{1}{n+2} + \delta\right) \\
&= \frac{2e^{2\delta}(1 + (n+2)\delta)}{(n+1)(n+2)e^{\delta n}n^{\delta}}
\end{aligned}$$

where the first inequality follows since $1 - x \leq e^{-x}$ for all x, and the second inequality since $\log n \leq \sum_{k=1}^{n} \frac{1}{n}$. Letting $\epsilon = \delta$, we have

$$\begin{aligned}
\mathbb{E}[\tau^{1+\delta}] &= \sum_{n=1}^{\infty} n^{1+\delta}\mathbb{P}(\tau = n) \\
&= \frac{1}{3} + \delta + \sum_{n=2}^{\infty}\frac{2e^{2\delta}n(1 + (n+2)\delta)}{e^{\delta n}(n+1)(n+2)} < \infty
\end{aligned}$$

By Theorem 2.2, the walk is diffusive, and the diffusion constant simplifies to

$$\nu = \frac{\mathbb{P}(\tau \text{ odd})}{\mathbb{P}(\tau \text{ even})\mathbb{E}[\tau]}.$$

It remains to show $\nu \to 0$ as $\delta \to 0^+$. To that end, recall $\mathbb{E}[\tau] = \sum_{n=1}^{\infty} \mathbb{P}(\tau \geq n)$. We have

$$\mathbb{P}(\tau \geq 1) = 1 \qquad \text{and} \qquad \mathbb{P}(\tau \geq n) = \prod_{k=2}^{n} \left(\frac{k}{k+1} - \delta \right)$$

for $n \geq 2$. The following computation gives a lower bound for $\mathbb{P}(\tau \geq n)$ when $n \geq 2$:

$$\begin{aligned}
\mathbb{P}(\tau \geq n) &= \prod_{k=2}^{n} \left(\frac{k}{k+1} - \delta \right) \\
&= \frac{2(1 - \frac{3\delta}{2})3(1 - \frac{4\delta}{3})...n(1 - \frac{(n+1)\delta}{n})}{3 \cdot 4... \cdot (n+1)} \\
&\geq \frac{2}{n+1} e^{-2(\frac{3\delta}{2})} e^{-2(\frac{4\delta}{3})} ... e^{-2(\frac{(n+1)\delta}{n})} \\
&= \frac{2}{n+1} \exp \left\{ -2 \sum_{k=2}^{n} \delta \left(1 + \frac{1}{k} \right) \right\} \\
&\geq \frac{2}{n+1} \exp \left\{ -2\delta(n - 2 + \log n + \gamma) \right\} \\
&= \frac{2e^{4\delta}}{(n+1)e^{2\delta\gamma} e^{2\delta n} n^{2\delta}} \\
&\geq \frac{2e^{4\delta}}{2ne^{2\delta\gamma} e^{2\delta n} n^{2\delta}}
\end{aligned}$$

where the first inequality follows since $1 - x \geq e^{-2x}$ holds for small $x \geq 0$, and the second equality since $\sum_{k=1}^{n} \frac{1}{k} \leq \log n + \gamma$, where γ is the Euler constant.

We will show $\sum_{n=1}^{\infty} \frac{2e^{4\delta}}{(n+1)e^{2\delta\gamma} e^{2\delta n} n^{2\delta}} \to \infty$ as $\delta \to 0^+$. Since the terms in the summation are positive and decreasing, we can use the integral test for convergence. After multiplying by some constant, it suffices to compute $\displaystyle\int_1^{\infty} \frac{e^{-2\delta x}}{x^{1+2\delta}} dx$.

Letting $t = -2\delta$, we have

$$\int_1^\infty \frac{e^{-2\delta x}}{x^{1+2\delta}} dx = \int_{2\delta}^\infty \frac{e^{-t}}{\left(\frac{t}{2\delta}\right)^{1+\delta}} \frac{dt}{2\delta} = (2\delta)^\delta \int_{2\delta}^\infty \frac{e^{-t}}{t^{1+2\delta}} dt = (2\delta)^\delta \Gamma(-2\delta, 2\delta)$$

where $\Gamma(\cdot, \cdot)$ is the Incomplete Upper Gamma function [1]. It is straightforward to verify that $(2\delta)^\delta \to 1$ as $\delta \to 0^+$. By [1], $\Gamma(-2\delta, 2\delta) \to \infty$ as $\delta \to 0^+$.

Thus, we have shown that a lower bound for $\mathbb{E}[\tau]$ diverges as δ tends to 0. By Theorem 2.2, ν converges to 0. Therefore the perturbed model is not strong enough to sustain a residual diffusivity.

2.4.2 Rate of Convergence

Since a residual diffusion is not achievable, it is natural to ask how fast ν is decreasing as δ tends to 0. In this section, we will verify that in the perturbed model, the diffusivity converges to 0 at a rate of $O\left(\frac{1}{|\log \delta|}\right)$.

Let $k = 2\delta$ and consider the integral above as a function of k:

$$f(k) = \int_1^\infty \frac{e^{-kx}}{x^{1+k}} dx \tag{2.4}$$

Then

$$f'(k) = -\int_1^\infty \frac{e^{-kx}}{x^{1+k}} (x + \log x) dx$$

Since $x \gg \log x$ as $x \to \infty$, $f'(k)$ is dominantly determined by the term with x, namely

$$f'(k) \sim -\int_1^\infty \frac{e^{-kx}}{x^k} dx$$

let $u = x^{1-k}$, so $du = (1-k)x^{-k}dx$, we have

$$f'(k) \sim -\frac{1}{1-k}\int_1^\infty e^{-ku^{\frac{1}{1-k}}}du = -\frac{1}{1-k}\int_1^\infty e^{-(k^{1-k}u)^{\frac{1}{1-k}}}du$$

let $v = k^{1-k}u$, the integral becomes

$$f'(k) \sim -\frac{k^{-1+k}}{1-k}\int_{k^{1-k}}^\infty e^{-v^{\frac{1}{1-k}}}dv$$

as $k \to 0^+$,

$$\int_{k^{1-k}}^\infty e^{-v^{\frac{1}{1-k}}}dv \to \int_0^\infty e^{-v}dv = 1$$

thus $f'(k) \sim -\frac{k^{-1+k}}{1-k}$ as $k \to 0^+$. Finally,

$$\lim_{k\to 0^+}\frac{-f(k)}{\log(\delta)} = \lim_{k\to 0^+}\frac{-f(k)}{\log k - \log 2} \overset{\text{L'H}}{=} \lim_{k\to 0^+}\frac{-f'(k)}{1/k} = \lim_{k\to 0^+}\frac{k^{-1+k}k}{1-k} = 1$$

An identical computation shows $\lim_{k\to 0^+}\frac{f(\delta)}{-\log\delta} = 1$. Since

$$\sum_{n=1}^\infty \frac{2e^{4\delta}}{(n+1)e^{2\delta\gamma}e^{2\delta n}n^{2\delta}} \leq \mathbb{E}[\tau] \leq \sum_{n=1}^\infty \frac{2e^{2\delta}}{(n+1)e^{\delta\gamma}e^{\delta n}n^{\delta}},$$

after multiplying by some constant, $\mathbb{E}[\tau] \sim C_1\,|\log\delta|$. Applying (2.2), $\nu_\delta = O\left(\frac{1}{|\log\delta|}\right)$.

2.4.3 Theorem 2.4: Existence of Positive Diffusion Constant

The formula for the diffusion constant ν follows directly from Theorem 2.2. The proof of Theorem 2.2 is based on the formula for the Green's function, and a standard Tauberian theorem. It utilized the following functions and quantities:

$$G_z(x) = \sum_{n=0}^{\infty} z^n \mathbb{P}(S_n = x), \quad \text{for } z \in [0,1]$$

$$\begin{cases} a_z = \displaystyle\sum_{n=2}^{\infty} z^n \mathbb{P}(\tau \geq n) \mathbb{1}_{\{n \text{ even}\}} \\ b_z = \displaystyle\sum_{n=2}^{\infty} z^n \mathbb{P}(\tau \geq n) \mathbb{1}_{\{n \text{ odd}\}} \end{cases} \qquad \begin{cases} p_z = \displaystyle\sum_{n=1}^{\infty} z^n \mathbb{P}(\tau = n) \mathbb{1}_{\{n \text{ even}\}} \\ q_z = \displaystyle\sum_{n=1}^{\infty} z^n \mathbb{P}(\tau = n) \mathbb{1}_{\{n \text{ odd}\}} \end{cases}$$

and other variables built up from a_z, b_z, p_z, and q_z. We will show below that, even though the model is stochastic, $\mathbb{P}(\tau \geq n)$ is still deterministic. Thus the proof of Theorem 2.2 still applies and gives the formula for ν.

Given that an edge has been traversed n^{th} times, let P_n denote the total probability of breaking out of this edge on the $(n+1)^{th}$ turn, and let Q_n denote the probability of traversing this edge again on the $(n+1)^{th}$ turn. Then loosely speaking, P_n is the sum of all the terms of the form $\frac{n+1}{n+2} - \delta\xi$, given that $\xi_n = \xi \leq \frac{n+1}{\delta(n+2)}$. Formally,

$$P_n = \int_0^{\frac{n+1}{\delta(n+2)}} \left(\frac{n+1}{n+2} - \delta x \right) f(x) dx$$

and

$$Q_n = \left(\int_0^{\frac{n+1}{\delta(n+2)}} \left(\frac{1}{n+2} - \delta x \right) f(x) dx \right) + \mathbb{P}\left(\xi_n > \frac{n+1}{\delta(n+2)} \right)$$

Similar to the previous result, for $n \geq 2$, we have

$$\mathbb{P}(\tau = n) = \left(\prod_{i=1}^{n-1} P_i \right) Q_n \qquad \text{and} \qquad \mathbb{P}(\tau \geq n) = \prod_{i=1}^{n-1} P_i.$$

An upper bound for $\mathbb{P}(\tau \geq n)$ is

$$\begin{aligned}
\mathbb{P}(\tau \geq n) &= \prod_{i=1}^{n-1}\left(\int_0^{\frac{i+1}{\delta(i+2)}}\left(\frac{i+1}{i+2}-\delta x\right)f(x)dx\right) \\
&\leq \prod_{i=1}^{n-1}\left(\frac{i+1}{i+2}-\delta\int_0^{\frac{i+1}{\delta(i+2)}}xf(x)dx\right) \\
&\leq \prod_{i=1}^{n-1}\left(\frac{i+1}{i+2}-\delta\int_0^{\frac{2}{3\delta}}xf(x)dx\right).
\end{aligned}$$

Let $\mu := \delta\int_0^{\frac{2}{3\delta}} xf(x)dx$. Then μ is a constant for each fixed δ. Thus $\mathbb{P}(\tau \geq n) = \prod_{i=1}^{n-1}\left(\frac{i+1}{i+2}-\mu\right)$, which has the same form as in the deterministic case. By a similar computation, there exists $\epsilon > 0$ such that $\mathbb{E}[\tau^{1+\epsilon}] < \infty$, and the walk is diffusive.

Recall Theorem 2.2, the diffusion constant is

$$\nu = \frac{\mathbb{P}(\tau \text{ odd})}{\mathbb{P}(\tau \text{ even})}\frac{1}{\mathbb{E}[\tau]}$$

In order to sustain residual diffusivity, we need $\mathbb{E}[\tau] \not\to \infty$ as $\delta \to 0^+$. Using the formula $\mathbb{E}[\tau] = \sum_{n=1}^{\infty}\mathbb{P}(\tau \geq n)$, we get

$$\mathbb{E}[\tau] = 1 + \sum_{n=2}^{\infty}\prod_{i=1}^{n-1}\left(\int_0^{\frac{i+1}{\delta(i+2)}}\left(\frac{i+1}{i+2}-\delta x\right)f(x)dx\right). \tag{2.5}$$

Suppose $\mathbb{E}[\xi_n] < \infty$. By Fatou's lemma,

$$\begin{aligned}
\liminf_{\delta \to 0^+} \mathbb{E}[\tau] &= \liminf_{\delta \to 0^+} \left(1 + \sum_{n=2}^{\infty} \left[\prod_{i=1}^{n-1} \int_0^{\frac{i+1}{\delta(i+2)}} \left(\frac{i+1}{i+2} - \delta x \right) f(x) dx \right] \right) \\
&\geq 1 + \sum_{n=2}^{\infty} \liminf_{\delta \to 0^+} \prod_{i=1}^{n-1} \left[\left(\frac{i+1}{i+2} \right) \int_0^{\frac{i+1}{\delta(i+2)}} f(x) dx - \delta \int_0^{\frac{i+1}{\delta(i+2)}} x f(x) dx \right] \\
&= 1 + \sum_{n=2}^{\infty} \liminf_{\delta \to 0^+} \left[\prod_{i=1}^{n-1} \left(\frac{i+1}{i+2} - \delta \mathbb{E}[\xi_n] \right) \right] \\
&= 1 + \sum_{n=2}^{\infty} \left(\prod_{i=1}^{n-1} \frac{i+1}{i+2} \right) \\
&= 1 + \sum_{n=2}^{\infty} \frac{2}{n+1} = \infty.
\end{aligned}$$

Since a lower bound for $\mathbb{E}[\tau]$ diverges to ∞, the corresponding upper bound for ν converges to 0. Thus $\nu \to 0$ as $\delta \to 0^+$. Moreover, since $\mathbb{E}[\xi_n]$ is a finite constant, the computation from Section 2.4.2 shows $\nu(\delta) = O(\frac{1}{|\log \delta|})$ as $\delta \to 0^+$.

2.4.4 Random Variables with Infinite Expectation

Necessary symptotic behavior of the pdf of ξ_n

Suppose ξ_n is a random variable with support in $[0, \infty)$ and $\mathbb{E}[\xi_n] = +\infty$. Let $f = f_{\xi_n}$ be the probability density function (pdf) of ξ_n, we have

$$\int_0^{\infty} f(x) dx = 1 \quad \text{and} \quad \int_0^{\infty} x f(x) dx = \infty.$$

We will study the asymptotic behavior of such f. Since $\int_0^{\infty} f(x) dx = 1$, we require $f(x) \leq O(x^{-n})$, for some $n > 1$.

On the other hand, $\int_0^{\infty} x f(x) dx = \infty$ implies $x f(x) \geq O(x^{-1})$. Thus, the necessary asymptotic behavior for f is

$$O\left(\frac{1}{x^2}\right) \leq f(x) < O\left(\frac{1}{x}\right).$$

Example 2.4.1. *A random variable ξ_n with $f(x) = O\left(\frac{1}{x^2}\right)$.*

Let ξ_n be non-negative Cauchy random variables with $x_0 = 0$ and pdf

$$f_{\xi_n}(x) = \frac{2}{\pi\gamma\left[1+\left(\frac{x}{\gamma}\right)^2\right]} = \frac{2\gamma}{\pi(x^2+\gamma^2)}.$$

Then

$$\begin{aligned}
\mathbb{P}(\tau \geq n) &= \prod_{i=1}^{n-1}\left[\left(\frac{i+1}{i+2}\right)\int_0^{\frac{i+1}{\delta(i+2)}} \frac{2\gamma}{\pi(x^2+\gamma^2)}dx - \delta\int_0^{\frac{i+1}{\delta(i+2)}} \frac{2\gamma x}{\pi(x^2+\gamma^2)}dx\right] \\
&= \prod_{i=1}^{n-1}\left[\left(\frac{i+1}{i+2}\right)\int_0^{\frac{i+1}{\delta(i+2)}} \frac{2\gamma}{\pi(x^2+\gamma^2)}dx - \delta\left(\frac{\gamma\log(x^2+\gamma^2)}{\pi}\right)\Bigg|_{x=0}^{x=\frac{i+1}{\delta(i+2)}}\right] \\
&= \prod_{i=1}^{n-1}\left[\left(\frac{i+1}{i+2}\right)\int_0^{\frac{i+1}{\delta(i+2)}} \frac{2\gamma}{\pi(x^2+\gamma^2)}dx - O\left(\delta\log\frac{1}{\delta}\right)\right]
\end{aligned}$$

and by Fatou's lemma,

$$\begin{aligned}
\liminf_{\delta\to 0^+}\mathbb{E}[\tau] &\geq 1 + \sum_{n=2}^{\infty}\liminf_{\delta\to 0^+}\mathbb{P}(\tau \geq n) \\
&= 1 + \sum_{n=2}^{\infty}\liminf_{\delta\to 0^+}\prod_{i=1}^{n-1}\left[\left(\frac{i+1}{i+2}\right)\int_0^{\frac{i+1}{\delta(i+2)}} \frac{2\gamma}{\pi(x^2+\gamma^2)}dx - O\left(\delta\log\frac{1}{\delta}\right).\right] \\
&= 1 + \sum_{n=2}^{\infty}\frac{2}{n+1} = \infty.
\end{aligned}$$

Similar to the previous section, since a lower bound for $\mathbb{E}[\tau]$ diverges to ∞, we have $\nu \to 0$ as $\delta \to 0^+$. Thus, even though the non-negative Cauchy distribution has a "fat" tail, the growth rate of $\int_0^{\frac{i+1}{\delta(i+2)}} xf(x)dx$ is still not fast enough to produce residual diffusivity.

Non-existence of residual diffusivity, rate of convergence

The case where $f(x) = O(x^{-2})$ was covered in example 2.4.1. In general, if

$$O\left(\frac{1}{x^2}\right) < f(x) < O\left(\frac{1}{x}\right)$$

then

$$O\left(\frac{1}{x}\right) < xf(x) < O(1)$$

which implies

$$\delta O\left(\log\frac{1}{\delta}\right) < \delta\int_0^{\frac{i+1}{\delta(i+2)}} xf(x) < \delta O\left(\frac{1}{\delta}\right).$$

Taking the limit as $\delta \to 0^+$, we have $\delta\int_0^{\frac{i+1}{\delta(i+2)}} xf(x) \to 0$, which implies

$$\mathbb{P}(\tau \geq n) = \prod_{i=1}^{n-1}\left(\int_0^{\frac{i+1}{\delta(i+2)}}\left(\frac{i+1}{i+2} - \delta x\right)f(x)dx\right) \to \frac{1}{n} \text{ as } \delta \to 0^+.$$

Therefore $\mathbb{E}[\tau] \to \infty$ and, subsequently, $\nu \to 0$.

For the asymptotic behavior of $\nu(\delta)$, we study 3 cases:

Case 1, $\boldsymbol{f(x) = O(x^{-2})}$:

By example 2.4.1, as $\delta \to 0^+$,

$$\mathbb{P}(\tau \geq n) = \prod_{i=1}^{n-1}\left(\frac{i+1}{i+2} - C\delta\log\left(\frac{1}{\delta}\right)\right)$$

A similar computation to the last part of section 2.4.1 shows that, after multiplying by some constant, to compute $\mathbb{E}[\tau]$, it suffices to compute

$$g(\delta\log(1/\delta)) = \int_1^\infty \frac{e^{-\delta\log(1/\delta)}}{x^{1+\delta\log(1/\delta)}}.$$

And by the computation of section 2.4.2, which shows $\lim_{\delta\to 0^+}\frac{g(k)}{\log k} = 1$, we have

$$\lim_{\delta\to 0}\frac{g(\delta\log(1/\delta))}{\log(\delta\log(1/\delta))} = 1.$$

This implies $\mathbb{E}[\tau] \sim C_1\log(|\delta\log\delta|)$, and therefore

$$\nu \sim \frac{C_2}{\log(|\delta\log\delta|)} \sim \frac{C_3}{\log\delta}.$$

Case 2, $\boldsymbol{f(x) = O(x^{-(1+j)})}$**, for** $\boldsymbol{0 < j < 1}$ **:**

A similar calculation to example 2.4.1 shows, as $\delta \to 0^+$,

$$\mathbb{P}(\tau \geq n) = \prod_{i=1}^{n-1}\left(\frac{i+1}{i+2} - C\delta^j\right)$$

and a calculation similar to Case 1 shows $\mathbb{E}[\tau] \sim C_1\,|\log(\delta^j)| = C_2\,|\log(\delta)|$. So in this case,

$$\nu \sim \frac{C_3}{|\log(\delta)|},$$

which gives the same result as the deterministic case.

Case 3, $\boldsymbol{f(x) < O(x^{-(1+j)})}$**, for any** $\boldsymbol{0 < j < 1}$ **and** $\boldsymbol{f(x) > O(x^{-2})}$ **:**

One such example is $f(x) = O\left(\frac{1}{x(\log x)^2}\right)$. Then

$$\int_0^{\frac{i+1}{\delta(i+2)}} xf(x)dx = \int_0^{\frac{i+1}{\delta(i+2)}} \frac{C}{\log^2 x}dx$$

which is a well known logarithm integral with asymptotic behavior:

$$\int \frac{1}{\log^2 x} dx = li(x) - \frac{x}{\log x} = O\left(\frac{x}{\log^2 x}\right)$$

therefore

$$\delta \int_0^{\frac{i+1}{\delta(i+2)}} xf(x)dx = \delta O\left(\frac{1}{\delta \log^2\left(\frac{C_1}{\delta}\right)}\right) = O\left(\frac{1}{\log^2(\delta)}\right)$$

as $\delta \to 0^+$. This implies

$$\mathbb{P}(\tau \geq n) = \prod_{i=1}^{n-1} \left(\frac{i+1}{i+2} - \frac{C_2}{\log^2 \delta}\right)$$

and a similar calculation to Case 1 shows $\mathbb{E}[\tau] \sim C_3 \log(\log^2 \delta) = C_4 \log|\log \delta|$. Thus, we have constructed a random variable ξ_n such that ν converges to zero at a rate of

$$\nu \sim \frac{C}{\log|\log \delta|}.$$

2.4.5 Proof of Theorem 2.5

Theorem 2.5 is a consequence of Theorem 2.4. The fact that the model is diffusive for any $\delta > 0$ follows directly. For the rate at which ν tends to 0, let f_n be the p.d.f. of ξ_n and recall that

$$\mathbb{P}(\tau \geq n) = \prod_{i=1}^{n-1} \left(\int_0^{\frac{i+1}{\delta(i+2)}} \left(\frac{i+1}{i+2} - \delta x\right) f_n(x)dx\right).$$

Since $\mathbb{E}[\xi_n] < \infty$ for all n, one can find a random variable Y with $\mathbb{E}[Y] = \infty$ with p.d.f. f_Y such that, for sufficiently small δ,

$$\delta \int_0^{\frac{i+1}{\delta(i+2)}} x f_n(x) dx \leq \delta \int_0^{\frac{i+1}{\delta(i+2)}} y f_Y(y) dy$$

so as $\delta \to 0^+$,

$$\mathbb{P}(\tau \geq n) \geq \prod_{i=1}^{n-1} \left(\frac{i+1}{i+2} - \delta \int_0^{\frac{i+1}{\delta(i+2)}} y f_Y(y) dy \right).$$

Notice the expression on the RHS matches the case of infinite expectation of the Theorem 2.4. Therefore $\mathbb{E}[\tau]$ grows at least as fast as the previous case, and hence so is the decay rate of ν_δ. One can choose Y so that $f_Y(y) = O(y^{-2})$ (Similar to Case 1 of section 2.4.4), so that $\nu_Y(\delta) \sim O(|\log \delta|)$. Then ν_δ decays at a rate of at most $O(|\log \delta|)$ (by section 2.4.3), and at least $O(\log \delta)$, from the previous case. It follows that $\nu_\delta = O(\log \delta)$.

2.4.6 Theorem 6: Results in Higher Dimensions

Perturbation under model I:

For $d \geq 2$, the model becomes

$$\mathbb{P}(S_{n+1} = S_n + x | \mathcal{F}_n) = \begin{cases} \max\{\frac{1+n}{2d+n} - \delta, 0\}, & \text{if } \{S_n, S_n + x\} = e_n, \\ \min\{\frac{1}{2d+n} + \delta, 1\}, & \text{if } \{S_n, S_n + x\} \neq e_n. \end{cases}$$

A similar computation to that of the one-dimensional case shows, for $n \geq 2d$,

$$\begin{aligned}
\mathbb{P}(\tau \geq n) &= \prod_{k=2}^{n} \left(\frac{1+k}{2d+k} - \delta \right) \\
&= \frac{(2d)!}{(n+2)(n+3)...(n+2d)} \left(1 - \frac{2d+1}{2}\delta\right) ... \left(1 - \frac{2d+n}{n+1}\delta\right) \\
&\to \frac{(2d)!}{(n+2)(n+3)...(n+2d)} \exp\left\{ -\delta \sum_{k=2}^{n} \left(1 + \frac{2d-1}{k}\right) \right\} \\
&\sim \frac{(2d)!}{(n+2)(n+3)...(n+2d)} \exp\left\{-\delta(n-1+(2d-1)\log n - (2d-1))\right\} \\
&= \frac{(2d)!}{(n+2)(n+3)...(n+2d)} \frac{e^{-\delta n} e^{2\delta d}}{n^{\delta(2d-1)}}
\end{aligned}$$

which has the same form as in the one-dimensional case. For $d \geq 2$, the unperturbed walk is diffusive, as $\sum_{n=1}^{\infty} \mathbb{P}(\tau \geq n) < \infty$. Let τ_δ denote the model perturbed by δ. By Dominated Convergence Theorem

$$\lim_{\delta \to 0^+} \mathbb{E}[\tau_\delta] = \lim_{\delta \to 0^+} \sum_{n=1}^{\infty} \mathbb{P}(\tau \geq n) = \sum_{n=1}^{\infty} \lim_{\delta \to 0^+} \mathbb{P}(\tau \geq n) = \mathbb{E}[\tau_0].$$

Thus $\nu_\delta \to \nu$ as $\delta \to 0^+$. For the enhanced diffusivity, by the integral test, it suffices to consider the integral $\int_1^\infty \frac{e^{-kx}}{x^{(2d-1)(1+k)}} dx =: f(k)$. We have

$$\frac{\partial}{\partial k} f(k) = \int_1^\infty \frac{e^{-kx}}{x^{(2d-1)k+2d-2}} (x + (2d-1)\log x) dx.$$

Since $d \geq 2$, the integral converges for any non-negative value of k. By the Dominated Convergence Theorem,

$$\lim_{k \to 0^+} \frac{\partial}{\partial k} f(k) = \int_1^\infty \lim_{k \to 0^+} \frac{e^{-kx}}{x^{(2d-1)k+2d-2}} (x + (2d-1)\log x) dx < \infty,$$

which implies that $\mathbb{E}[\tau_\delta]$ grows at a linear rate near $\delta = 0$, and therefore

$$\nu_\delta = \nu_0 + O(\delta).$$

Perturbation under model II and III:

Let us consider the model

$$\mathbb{P}(S_{n+1} = S_n + x | \mathcal{F}_n) = \begin{cases} \max\{\frac{1+n}{2d+n} - \delta\xi_n, 0\}, & \text{if } \{S_n, S_n + x\} = e_n, \\ \min\{\frac{1}{2d+n} + \delta\xi_n, 1\}, & \text{if } \{S_n, S_n + x\} \neq e_n. \end{cases}$$

If $(\xi_n)_{n\in\mathbb{N}}$ is a sequence of random variables such that $\mathbb{E}[\xi_n] < \infty$, for all n, one can use an analogous argument to that of section 2.4.3 to show $\nu_\delta \to \nu_0$ at the same rate as model I. When $\mathbb{E}[\xi_n] = \infty$, let $f = f_{\xi_n}$. The proof of all three cases are identical. We present the proof of the second case below:

Case 2: $\boldsymbol{f(x) = O(x^{-(1+j)})}$**, for** $\boldsymbol{0 < j < 1}$ **:**

Using a similar computation to section 2.4.4, we have

$$\begin{aligned} \mathbb{P}(\tau \geq n) &= \prod_{k=2}^{n} \left(\int_0^{\frac{1+k}{\delta(2d+k)}} \left(\frac{1+k}{2d+k} - \delta x \right) f(x) dx \right) \\ &\to \prod_{k=2}^{n} \left(\frac{1+k}{2d+k} - C_1 \delta^j \right) \\ &\to \frac{C_2}{(n+2)(n+3)...(n+2d)} \frac{e^{-\delta^j n} e^{2\delta^j d}}{n^{\delta^j (2d-1)}} \end{aligned}$$

and the Dominated Convergence Theorem guarantees convergence of ν_δ. For the enhanced diffusivity, it suffices to consider the integral $\int_1^\infty \frac{e^{-k^j x}}{x^{(2d-1)(1+k^j)}} dx =: f(k^j)$. In this case,

$$\frac{\partial}{\partial k^j} f(k^j) = \int_1^\infty \frac{e^{-k^j x}}{x^{(2d-1)(1+k^j)}} (x + (2d-1) \log x) dx < \infty$$

the integral converges for any non-negative value of k. This implies $\mathbb{E}[\tau_\delta]$ grows at a rate of δ^j near $\delta = 0$. Therefore

$$\nu_\delta = \nu_0 + O(\delta^j).$$

Using an analogous argument, one gets the result for Cases 1 and 3, where the construction for Case 3 is the same as in section 2.4.4.

Chapter 3

The Relaxed Variable Splitting Method

3.1 Preliminaries

3.1.1 The One-layer Non-overlap Network

Consider a one-layer non-overlap network (Figure 1.1). For this model, a filter $\boldsymbol{w} \in \mathbb{R}^d$ is shared among k different hidden nodes, and the input feature is $\boldsymbol{x} \in \mathbb{R}^{kd}$. We assume the filter $\boldsymbol{w}$ is applied in a non-overlap way to k patches of the input: $\boldsymbol{x}[1], ..., \boldsymbol{x}[k]$, each with size d. We also assume that the input vectors $\boldsymbol{x}$ are i.i.d. Gaussian random vectors with zero mean and unit variance, and let $\mathcal{G}$ denote this distribution. Let $L(\boldsymbol{x}, \boldsymbol{w}) := \frac{1}{k}\sum_{i=1}^{k}\sigma(\boldsymbol{w} \cdot \boldsymbol{x}[i])$ be the output of this network. We assume there exists a ground truth $\boldsymbol{w}^*$ by which the training data is generated. The population risk is then:

$$f(\boldsymbol{w}) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{G}}[(L(\boldsymbol{x}; \boldsymbol{w}) - L(\boldsymbol{x}; \boldsymbol{w}^*))^2].$$

We define

$$g(\boldsymbol{u},\boldsymbol{v}) = \mathbb{E}_{\mathcal{G}}[\sigma(\boldsymbol{u}\cdot\boldsymbol{x})\sigma(\boldsymbol{v}\cdot\boldsymbol{x})]. \tag{3.1}$$

Lemma 1. *[7, 13] Assume $\boldsymbol{x}\in\mathbb{R}^d$ is a vector where the entries are i.i.d. Gaussian random variables with mean 0 and variance 1. Given $\boldsymbol{u},\boldsymbol{v}\in\mathbb{R}^d$, denote by $\theta_{\boldsymbol{u},\boldsymbol{v}}$ the angle between $\boldsymbol{u}$ and $\boldsymbol{v}$. Then*

$$g(\boldsymbol{u},\boldsymbol{v}) = \frac{1}{2\pi}\|\boldsymbol{u}\|\|\boldsymbol{v}\|\left(\sin\theta_{\boldsymbol{u},\boldsymbol{v}} + (\pi-\theta_{\boldsymbol{u},\boldsymbol{v}})\cos\theta_{\boldsymbol{u},\boldsymbol{v}}\right).$$

For the no-overlap network, the population risk is simplified to:

$$f(\boldsymbol{w}) = \frac{1}{k^2}\left[a(\|\boldsymbol{w}\|^2+\|\boldsymbol{w}^*\|^2) - 2kg(\boldsymbol{w},\boldsymbol{w}^*) - 2b\|\boldsymbol{w}\|\|\boldsymbol{w}^*\|\right]. \tag{3.2}$$

where $b=\frac{k^2-k}{2\pi}$ and $a=b+\frac{k}{2}$.

3.1.2 Modification with Binarized Activation

The model described here is identical to the section above, with the exception of the activation function. We honor the original notation of [73] and give a brief summary of the structure, as well as the associated loss function and gradient. Consider a one-layer non-overlap network with input $\boldsymbol{Z}\in\mathbb{R}^{k\times d}$ and filter $\boldsymbol{w}\in\mathbb{R}^d$. Let σ denote the binarized ReLU activation function, $\sigma(z):=\chi_{\{z>0\}}$. The empirical risk for each input $\boldsymbol{Z}$ is then

$$l(\boldsymbol{w},\boldsymbol{Z}) := (\mathbf{1}^T\sigma(\boldsymbol{Z}\boldsymbol{w}) - \mathbf{1}^T\sigma(\boldsymbol{Z}\boldsymbol{w}^*))^2, \tag{3.3}$$

where $\boldsymbol{w}^*\in\mathbb{R}^d$ is the underlying (non-zero) teaching parameter. Note that (3.4) is invariant under scaling $\boldsymbol{w}\to\boldsymbol{w}/c$, $\boldsymbol{w}^*\to\boldsymbol{w}^*/c$, for any scalar $c>0$. Without loss of generality, we assume $\|\boldsymbol{w}^*\| = 1$. Similar to the previous section, we are interested in minimizing the population loss:

$$f(\boldsymbol{w}) := \mathbb{E}_{\boldsymbol{Z}\sim\mathcal{G}}[(\mathbf{1}^T\sigma(\boldsymbol{Z}\boldsymbol{w}) - \mathbf{1}^T\sigma(\boldsymbol{Z}\boldsymbol{w}^*))^2] \tag{3.4}$$

The empirical risk function in (3.3) is piece-wise constant and has a.e. zero partial $\boldsymbol{w}$ gradient. If σ were differentiable, the back-propagation would read:

$$\frac{\partial\, l}{\partial \boldsymbol{w}}(\boldsymbol{w}, \boldsymbol{Z}) = \boldsymbol{Z}^T\sigma'(\boldsymbol{Z}\boldsymbol{w})(\sigma(\boldsymbol{Z}\boldsymbol{w}) - \sigma(\boldsymbol{Z}\boldsymbol{w}^*)). \tag{3.5}$$

However, σ has zero derivative a.e., rendering (3.5) inapplicable. We study the coarse gradient descent with σ' in (3.5) replaced by the (sub)derivative μ' of the regular ReLU function $\mu(x) := \max(x, 0)$. More precisely, we use the following surrogate of $\frac{\partial l}{\partial \boldsymbol{w}}(\boldsymbol{w}, \boldsymbol{Z})$:

$$g(\boldsymbol{w}, \boldsymbol{Z}) = \sqrt{\frac{2}{\pi}}\boldsymbol{Z}^T\mu'(\boldsymbol{Z}\boldsymbol{w})(\sigma(\boldsymbol{Z}\boldsymbol{w}) - \sigma(\boldsymbol{Z}\boldsymbol{w}^*)) \tag{3.6}$$

with $\mu'(x) = \sigma(x)$. The constant $\sqrt{\frac{2}{\pi}}$ represents a ReLU function μ with a smaller slope, and will be necessary to give a stronger convergence result for our main findings. To simplify our analysis, we let $N \uparrow \infty$ in (3.3), so that its coarse gradient approaches $\mathbb{E}_{\boldsymbol{Z}}[g(\boldsymbol{w}, \boldsymbol{Z})]$. The following lemma asserts that $\mathbb{E}_{\boldsymbol{Z}}[g(\boldsymbol{w}, \boldsymbol{Z})]$ has positive correlation with the true gradient $\nabla f(\boldsymbol{w})$, and consequently, $-\mathbb{E}_{\boldsymbol{Z}}[g(\boldsymbol{w}, \boldsymbol{Z})]$ gives a reasonable descent direction.

Lemma 2. *[73] If $\theta(\boldsymbol{w}, \boldsymbol{w}^*) \in (0, \pi)$, and $\|\boldsymbol{w}\| \neq 0$, then the inner product between the expected coarse and true gradient w.r.t. $\boldsymbol{w}$ is*

$$\langle \mathbb{E}_{\boldsymbol{Z}}[g(\boldsymbol{w}, \boldsymbol{Z})], \nabla f(\boldsymbol{w})\rangle = \frac{\sin(\theta(\boldsymbol{w}, \boldsymbol{w}^*))}{2(\sqrt{2\pi})^3\|\boldsymbol{w}\|}k^2 \geq 0.$$

3.1.3 The Relaxed Variable Splitting Method

Let $\eta > 0$ denote the training step size. Consider a simple gradient descent update:

$$\boldsymbol{w}^{t+1} = \boldsymbol{w}^t - \eta \nabla f(\boldsymbol{w}^t). \tag{3.7}$$

It was shown [7] that the one-layer non-overlap network can be learned with high probability and in polynomial time. We seek to improve sparsity in the limit weight while also maintain good accuracy. A classical method to accomplish this task is to introduce ℓ_1 regularization to the population loss function, and the modified gradient update rule. Consider the minimization problem:

$$l(\boldsymbol{w}) = f(\boldsymbol{w}) + \lambda \|\boldsymbol{w}\|_1. \tag{3.8}$$

for some $\lambda > 0$. We propose a new approach to solve this minimization problem, called the Relaxed Variable Splitting Method (RVSM). We first convert (3.8) into an equation of two variables

$$l(\boldsymbol{w}, \boldsymbol{u}) = f(\boldsymbol{w}) + \lambda \|\boldsymbol{u}\|_1.$$

and consider the augmented Lagrangian

$$\mathcal{L}_\beta(\boldsymbol{w}, \boldsymbol{u}) = f(\boldsymbol{w}) + \lambda \|\boldsymbol{u}\|_1 + \frac{\beta}{2}\|\boldsymbol{w} - \boldsymbol{u}\|^2. \tag{3.9}$$

We minimize $\mathcal{L}_\beta(\boldsymbol{w}^t, \boldsymbol{u}^t)$ by alternatively decreasing the Lagrangian in $\boldsymbol{w}^t$ and $\boldsymbol{u}^t$: the update on $\boldsymbol{w}^t$ is a simple gradient descent step, and the update on $\boldsymbol{u}^t$ is $\boldsymbol{u}^{t+1} = \arg\min_{\boldsymbol{u}} \mathcal{L}_\beta(\boldsymbol{w}^t, \boldsymbol{u})$. Let $S_{\lambda/\beta}(\boldsymbol{w}) := sgn(\boldsymbol{w})(|\boldsymbol{w}| - \lambda/\beta)\chi_{\{|\boldsymbol{w}|>\lambda/\beta\}}$ be the soft thresholding operator. The RVSM is described in Algorithm 1; and the variation for binarized activation, RVSCGD, is described in Algorithm 2.

Algorithm 1 RVSM

```
Input: $\eta, \beta, \lambda$, $max_{epoch}$, $max_{batch}$
Initialize: $\boldsymbol{w}^0$
Define: $\boldsymbol{u}^0 = S_{\lambda/\beta}(\boldsymbol{w}^0)$
for $t = 0, 1, 2, ..., max_{epoch}$ do
  for $batch = 1, 2, ..., max_{batch}$ do
    $\boldsymbol{w}^{t+1} \leftarrow \boldsymbol{w}^t - \eta \nabla f(\boldsymbol{w}^t) - \eta\beta(\boldsymbol{w}^t - \boldsymbol{u}^t)$
    $\boldsymbol{u}^{t+1} \leftarrow \arg\min_{\boldsymbol{u}} \mathcal{L}_\beta(\boldsymbol{w}^t, \boldsymbol{u}) = S_{\lambda/\beta}(\boldsymbol{w}^t)$
  end for
end for
Output: $\boldsymbol{u}^t, \boldsymbol{w}^t$
```

Algorithm 2 RVSCGD

```
Input: $\eta, \beta, \lambda$, $max_{epoch}$, $max_{batch}$
Initialize: $\boldsymbol{w}^0$
Define: $\boldsymbol{u}^0 = S_{\lambda/\beta}(\boldsymbol{w}^0)$
for $t = 0, 1, 2, ..., max_{epoch}$ do
  for $batch = 1, 2, ..., max_{batch}$ do
    $\hat{\boldsymbol{w}}^{t+1} \leftarrow \boldsymbol{w}^t - \eta \mathbb{E}_{\boldsymbol{Z}}[g(\boldsymbol{w}^t, \boldsymbol{Z})] - \eta\beta(\boldsymbol{w}^t - \boldsymbol{u}^{t+1})$
    $\boldsymbol{w}^{t+1} = \frac{\hat{\boldsymbol{w}}^{t+1}}{\|\hat{\boldsymbol{w}}^{t+1}\|}$
    $\boldsymbol{u}^{t+1} \leftarrow \arg\min_{\boldsymbol{u}} \mathcal{L}_\beta(\boldsymbol{w}^t, \boldsymbol{u}) = S_{\lambda/\beta}(\boldsymbol{w}^t)$
  end for
end for
Output: $\boldsymbol{u}^t, \boldsymbol{w}^t$
```

3.1.4 Comparison with ADMM

A well-known, modern classical to solve the minimization problem (3.8) is the Alternating Direction Method of Multipliers (or ADMM). In ADMM, we consider the Lagrangian

$$\mathcal{L}_\beta(\boldsymbol{w}, \boldsymbol{u}, \boldsymbol{z}) = f(\boldsymbol{w}) + \lambda \|\boldsymbol{u}\|_1 + \langle \boldsymbol{z}, \boldsymbol{w} - \boldsymbol{u} \rangle + \frac{\beta}{2}\|\boldsymbol{w} - \boldsymbol{u}\|^2. \tag{3.10}$$

and apply the updates:

$$\begin{cases} \boldsymbol{w}^{t+1} \leftarrow \arg\min_{\boldsymbol{w}} \mathcal{L}_\beta(\boldsymbol{w}, \boldsymbol{u}^t, \boldsymbol{z}^t) \\ \boldsymbol{u}^{t+1} \leftarrow \arg\min_{\boldsymbol{u}} \mathcal{L}_\beta(\boldsymbol{w}^{t+1}, \boldsymbol{u}, \boldsymbol{z}^t) \\ \boldsymbol{z}^{t+1} \leftarrow \boldsymbol{z}^t + \beta(\boldsymbol{w}^{t+1} - \boldsymbol{u}^{t+1}) \end{cases} \tag{3.11}$$

Although widely used in practice, the ADMM method has several drawbacks when it comes to regularizing deep neural networks: First, the loss function f is often non-convex and only differentiable in some very specific regions, thus the current theory of optimizations does not apply [67]. Secondly, the update

$$\boldsymbol{w}^{t+1} \leftarrow \arg\min_{\boldsymbol{w}} \mathcal{L}_\beta(\boldsymbol{w}^{t+1}, \boldsymbol{u}, \boldsymbol{z}^t)$$

is not applicable in practice, as it requires one to know fully how $f(\boldsymbol{w})$ behaves. In most ADMM adaptations on DNN, this step is replaced by a simple gradient descent. Lastly, the Lagrange multiplier $\boldsymbol{z}^t$ tends to reduce the sparsity of the limit of $\boldsymbol{u}^t$, as it seeks to close the gap between $\boldsymbol{w}^t$ and $\boldsymbol{u}^t$.

In contrast, the RVSM method resolves all these difficulties presented by ADMM. First, we will show that in a one-layer non-overlap network, the iterations will keep $\boldsymbol{w}^t$ and $\boldsymbol{u}^t$ in a nice region, where one can guarantee Lipschitz gradient property for $f(\boldsymbol{w})$. Secondly, the update of $\boldsymbol{w}^t$ is not an $\arg\min$ update, but rather a gradient descent iteration itself, so our theory does not deviate from practice. Lastly, without the Lagrange multiplier term $\boldsymbol{z}^t$, there will be a gap between $\boldsymbol{w}^t$ and $\boldsymbol{u}^t$ at the limit. The $\boldsymbol{u}^t$ is much more sparse than in the case of ADMM, and numerical results shows that one can replace $\boldsymbol{w}^t$ by $\boldsymbol{u}^t$ after each training epoch without incurring any performance loss. An intuitive explanation for this is that when the dimension of $\boldsymbol{w}^t$ is high, most of its components that will be pruned off to get $\boldsymbol{u}^t$ have very small magnitudes, and are often the redundant weights.

In short, the RVSM method is easier to implement (no need to keep track of the variable $\boldsymbol{z}^t$), can greatly increase sparsity in the weight variable $\boldsymbol{u}^t$, while also maintaining the same performance as ADMM. Moreover, RVSM has convergence guarantee and limit characterization as stated below.

3.2 Convergence Results

3.2.1 The One-layer Model with ReLU Activation

Before we state our main results, the following Lemma is needed to establish the existence of a Lipschitz constant L:

Lemma 3. *(Lipschitz gradient)*
There exists a global constant L such that the iterations of Algorithm 1 satisfy

$$\|\nabla f(\boldsymbol{w}^t) - \nabla f(\boldsymbol{w}^{t+1})\| \leq L\|\boldsymbol{w}^t - \boldsymbol{w}^{t+1}\|, \;\; \forall t. \tag{3.12}$$

An important consequence of Lemma 3 is: for all t, the iterations of Algorithm 1 satisfy:

$$f(\boldsymbol{w}^{t+1}) - f(\boldsymbol{w}^t) \leq \langle \nabla f(\boldsymbol{w}^t), \boldsymbol{w}^{t+1} - \boldsymbol{w}^t \rangle + \frac{L}{2}\|\boldsymbol{w}^{t+1} - \boldsymbol{w}^t\|^2.$$

Theorem 3.1. *Suppose the initialization of the RVSM Algorithm satisfies:*

(i) Step size η is small so that $\eta \leq \frac{1}{\beta+L}$;

(ii) Initial angle $\theta(\boldsymbol{w}^0, \boldsymbol{w}^*) \leq \pi - \delta$, for some $\delta > 0$;

(iii) Parameters k, β, λ are such that $k \geq 2, \beta \leq \frac{\delta \sin \delta}{k\pi}$, and $\frac{\lambda}{\beta} < \frac{1}{\sqrt{d}}$.

Then the Lagrangian $\mathcal{L}_\beta(\boldsymbol{w}^t, \boldsymbol{u}^t)$ decreases monotonically; and $(\boldsymbol{w}^t, \boldsymbol{u}^t)$ converges sub-sequentially to a limit point $(\bar{\boldsymbol{w}}, \bar{\boldsymbol{u}})$, with $\bar{\boldsymbol{u}} = S_{\lambda/\beta}(\bar{\boldsymbol{w}})$, such that:

(i) $0 \in \partial_{\boldsymbol{u}} \mathcal{L}_\beta(\bar{\boldsymbol{w}}, \bar{\boldsymbol{u}})$ and $\nabla_{\boldsymbol{w}} \mathcal{L}_\beta(\bar{\boldsymbol{w}}, \bar{\boldsymbol{u}}) = 0$;

(ii) $\nabla_{\boldsymbol{w}} \mathcal{L}_\beta(\boldsymbol{w}^t, \boldsymbol{u}^t) = O(\epsilon)$ in $O(1/\epsilon^2)$ iterations;

(iii) The limit point $\bar{\boldsymbol{w}}$ is close to the ground truth $\boldsymbol{w}^*$ in the sense that $\theta(\bar{\boldsymbol{w}}, \boldsymbol{w}^*) < \delta$ and $\|\bar{\boldsymbol{w}} - \boldsymbol{w}^*\| = O(\beta)$.

The full proof of Theorem 3.1 is given in the next section. Here we overview the key steps. First, we show that the iterations of Algorithm 1 will eventually bring $\boldsymbol{w}^t$ to within a closed annulus D of width $2\,M$ around the sphere centered at origin with radius $\|\boldsymbol{w}^*\|$. In other words, there exists a T such that for all $t \geq T, \|\boldsymbol{w}^t\| \in [\|\boldsymbol{w}^*\| - M, \|\boldsymbol{w}^*\| + M]$, for some $0 < M < \|\boldsymbol{w}^*\|$. Then with no loss of generality, we can assume that $\boldsymbol{w}^t$ is in this closed strip, for all t.

Next, for the region D of the iterations, we will show there exists a global constant L such that the Lipschitz gradient property in Lemma 3 holds.

Finally, the Lipschitz gradient property allows us to show the descent of angle θ^t and Lagrangian $\mathcal{L}_\beta(\boldsymbol{w}^t, \boldsymbol{u}^t)$. The conditions on η, β, λ are used to show $\theta^{t+1} \leq \theta^t$; and an analysis of the limit point gives the bound on $\theta(\bar{\boldsymbol{w}}, \boldsymbol{w}^*)$ and $\|\bar{\boldsymbol{w}} - \boldsymbol{w}^*\|$. From the descent property of $\mathcal{L}_\beta(\boldsymbol{w}^t, \boldsymbol{u}^t)$, classical results from optimization [7] can be used to show that after $T = O\left(\epsilon^{-2}\right)$ iterations, we have $\nabla_{\boldsymbol{w}} \mathcal{L}_\beta(\boldsymbol{w}^t, \boldsymbol{u}^t) = O(\epsilon)$, for some $t \in (0, T]$. This leads to the desired convergence rate and finishes the proof.

It should be noted that without the condition on β being small, one may not guarantee monotonicity of θ^t. However, it still can be shown that $\mathcal{L}_\beta(\boldsymbol{w}^t, \boldsymbol{u}^t)$ decreases and thus the iteration will converge to some limit point $(\bar{\boldsymbol{w}}, \bar{\boldsymbol{u}})$. In this case, the limit point may not be near the ground truth $\boldsymbol{w}^*$; i.e. we may not have $\theta(\bar{\boldsymbol{w}}, \boldsymbol{w}^*) < \delta$. Furthermore, the bound on $\|\bar{\boldsymbol{w}} - \boldsymbol{w}^*\|$ will also be weaker.

Corollary 2. *Suppose the initialization of the RVSM Algorithm satisfies Theorem 3.1, then the $\bar{\boldsymbol{w}}$ equation below holds:*

$$\boldsymbol{w}^* = \frac{k\pi}{\pi - \theta}\beta(\bar{\boldsymbol{w}} - S_{\lambda/\beta}(\bar{\boldsymbol{w}})) + C\bar{\boldsymbol{w}}, \tag{3.13}$$

where $\theta := \theta(\bar{\boldsymbol{w}}, \boldsymbol{w}^*)$, constant $C \in (0, \frac{1}{1-2k\lambda\sqrt{d}})$. Since component-wise, $\bar{\boldsymbol{w}} - S_{\lambda/\beta}(\bar{\boldsymbol{w}})$ has the same sign as $\bar{\boldsymbol{w}}$, the ground truth $\boldsymbol{w}^*$ is an expansion of $C\,\bar{\boldsymbol{w}}$, or equivalently $\bar{\boldsymbol{w}}$ is (up to scalar multiple) a shrinkage of $\boldsymbol{w}^*$.

The proofs of Theorem 1 and Corollary 1.1 do not require convexity of the regularization term $\lambda\|\boldsymbol{u}\|_1$, hence extend to other sparse penalties such as ℓ_0 and transformed ℓ_1 penalty [76]. We have:

Corollary 3. *Under the conditions of Theorem 1 however with the l_1 penalty replaced by an ℓ_0 or transformed-ℓ_1 penalty, the RVSM Algorithm converges sub-sequentially to a limit point $(\bar{\boldsymbol{w}}, \bar{\boldsymbol{u}})$ satisfying $\nabla_{\boldsymbol{w}}\mathcal{L}_\beta(\bar{\boldsymbol{w}}, \bar{\boldsymbol{u}}) = 0$. The Lagrangian and angle θ^t also decrease monotonically, with the limit angle satisfying $\theta(\bar{\boldsymbol{w}}, \boldsymbol{w}^*) < \delta$. Here $\bar{\boldsymbol{u}}$ is a thresholding of $\bar{\boldsymbol{w}}$, and equation (3.13) holds with $S_{\lambda/\beta}(\cdot)$ replaced by the thresholding operator of the corresponding penalty.*

3.2.2 The One-layer Model with Binarized ReLU Activation

Theorem 3.2. *Suppose that the initialization and penalty parameters of the RVSCGD algorithm satisfy:*

(i) $\theta(\boldsymbol{w}^0, \boldsymbol{w}^*) \leq \pi - \delta$*, for some* $\delta > 0$*;*

(ii) $\beta \leq \frac{k \sin\delta}{2\sqrt{2\pi}}$*, and* $\lambda < \frac{k}{2\sqrt{2\pi d}}$*;*

(iii) η *is small such* $\eta \leq \min\left\{\frac{1}{\beta+L}, \frac{2\sqrt{2\pi}}{k}\right\}$*, where* L *is the Lipschitz constant in Lemma 10; and for all* t*,* $\eta \left\|\mathbb{E}_{\boldsymbol{Z}}[g(\boldsymbol{w}^t, \boldsymbol{Z})] + \beta\left(\boldsymbol{w}^t - \boldsymbol{u}^{t+1}\right)\right\| \leq \frac{1}{2}$*.*

Then the Lagrangian $\mathcal{L}_\beta(\boldsymbol{u}^t, \boldsymbol{w}^t)$ *decreases monotonically; and* $(\boldsymbol{u}^t, \boldsymbol{w}^t)$ *converges sub-sequentially to a limit point* $(\bar{\boldsymbol{u}}, \bar{\boldsymbol{w}})$*, with* $\bar{\boldsymbol{u}} = S_{\lambda/\beta}(\bar{\boldsymbol{w}})$*, such that:*

(i) Let $\theta := \theta(\bar{\boldsymbol{w}}, \boldsymbol{w}^*)$ *and* $\gamma := \theta(\bar{\boldsymbol{u}}, \bar{\boldsymbol{w}})$*, then* $\theta < \delta$

(ii) The limit point $(\bar{\boldsymbol{u}}, \bar{\boldsymbol{w}})$ *satisfies* $\bar{\boldsymbol{u}} = S_{\lambda/\beta}(\bar{\boldsymbol{w}})$ *and*

$$\boldsymbol{w}^* = \frac{2\sqrt{2\pi}}{k}\beta(\bar{\boldsymbol{w}} - S_{\lambda/\beta}(\bar{\boldsymbol{w}})) + C\bar{\boldsymbol{w}} \tag{3.14}$$

where $S_{\lambda/\beta}(\cdot)$ *is the soft-thresholding operator of* ℓ_1*, for some constant* $0 < C \leq \frac{k}{k - 2\lambda\sqrt{2\pi d}}$

(iii) The limit point $\bar{\boldsymbol{w}}$ *is close to the ground truth* $\boldsymbol{w}^*$ *such that*

$$\|\boldsymbol{w}^* - \bar{\boldsymbol{w}}\,\| \leq \frac{4\sqrt{2\pi}\beta\sin\gamma}{k}. \tag{3.15}$$

Remark. *As the sign of* $(\bar{\boldsymbol{w}} - S_{\lambda/\beta}(\bar{\boldsymbol{w}}))$ *agrees with* $\bar{\boldsymbol{w}}$*, eq. (3.14) implies that* $\boldsymbol{w}^*$ *equals an expansion of* $C\,\bar{\boldsymbol{w}}$ *or equivalently* $\bar{\boldsymbol{w}}$ *is (up to a scalar multiple) a shrinkage of* w^**, which explains the source of sparsity in* $\bar{\mathbf{w}}$*. The assumption on* η *is reasonable, as will be shown below:* $\|\mathbb{E}_{\boldsymbol{Z}}[g(\boldsymbol{w}^t, \boldsymbol{Z})]\|$ *is bounded away from zero, and thus* $\|\mathbb{E}_{\boldsymbol{Z}}[g(\boldsymbol{w}^t, \boldsymbol{Z})] + \beta(\boldsymbol{w}^t - \boldsymbol{u}^{t+1})\|$ *is also bounded.*

The proof is provided in details in section 3.4. Here we provide an overview of the key steps. First, we show that there exists a constant L_f such that

$$\|\nabla f(\boldsymbol{w}^{t+1}) - \nabla f(\boldsymbol{w}^t)\| \leq L_f\|\boldsymbol{w}^{t+1} - \boldsymbol{w}^t\|$$

then we show that the Lipschitz gradient property still holds when replaced by the coarse gradient:

$$\|\mathbb{E}_{\boldsymbol{Z}}[g(\boldsymbol{w}^{t+1},\boldsymbol{Z})]-\mathbb{E}_{\boldsymbol{Z}}[g(\boldsymbol{w}^{t},\boldsymbol{Z})]\|\leq K\|\boldsymbol{w}^{t+1}-\boldsymbol{w}^{t}\|$$

and subsequently show

$$f(\boldsymbol{w}_2)-f(\boldsymbol{w}_1)\leq\langle\nabla\mathbb{E}_{\boldsymbol{Z}}[g(\boldsymbol{w}_1,\boldsymbol{Z})],\boldsymbol{w}_2-\boldsymbol{w}_1\rangle+\frac{L}{2}\|\boldsymbol{w}_2-\boldsymbol{w}_1\|^2.$$

These inequalities hold when $\|\boldsymbol{w}^t\|\geq M$, for some $M>0$. It can be shown that with bad initialization, one may have $\|\boldsymbol{w}^t\|\to 0$ as $t\to\infty$. We circumvent this problem by normalizing $\boldsymbol{w}^t$ at each iteration.

Next, we show the iterations satisfy $\theta^{t+1}\leq\theta^t$, and $\mathcal{L}_\beta(\boldsymbol{u}^{t+1},\boldsymbol{w}^{t+1})\leq\mathcal{L}_\beta(\boldsymbol{u}^t,\boldsymbol{w}^t)$. Finally, an analysis of the stationary point yields the desired bound.

In none of these steps do we use convexity of the ℓ_1 penalty term. Here we extend our result to ℓ_0 and transformed ℓ_1 (Tℓ_1) regularization [77].

Corollary 4. *Suppose that the initialization of the RVSCGD algorithm satisfies the conditions in Theorem 3.2, and that the ℓ_1 penalty is replaced by ℓ_0 or Tℓ_1. Then the RVSCGD iterations converge to a limit point $(\bar{\boldsymbol{u}},\bar{\boldsymbol{w}})$ satisfying equation* (3.14) *with ℓ_0's hard thresholding operator [5] or Tℓ_1 thresholding [76] replacing $S_{\lambda/\beta}$, and similar bound* (3.15) *holds.*

3.3 Proof of the First Convergence Result

The following Lemmas will be needed to prove Theorem 3.1:

Lemma 4. *(Properties of the gradient, [7])*

For the loss function $f(\boldsymbol{w})$ of equation (3.2), the following holds:

1. $f(\boldsymbol{w})$ is differentiable if and only if $\mathbf{w}\neq 0$.

2. For $k > 1, f(\boldsymbol{w})$ has three critical points:

(a) A local maximum at $\boldsymbol{w} = 0$.

(b) A unique global minimum at $\boldsymbol{w} = \boldsymbol{w}^*$.

(c) A degenerate saddle point at $\boldsymbol{w} = -\left(\frac{k^2-k}{k^2+(\pi-1)k}\right)\boldsymbol{w}^*$.

For $k = 1, w = 0$ is not a local maximum and the unique global minimum $\boldsymbol{w}^*$ is the only differentiable critical point.

Given $\theta := \theta(\boldsymbol{w}, \boldsymbol{w}^*)$, the gradient of f can be expressed as

$$\nabla f(\boldsymbol{w}) = \frac{1}{k^2}\left[\left(k + \frac{k^2-k}{\pi} - \frac{k}{\pi}\frac{\|\boldsymbol{w}^*\|}{\|\boldsymbol{w}\|}\sin\theta - \frac{k^2-k}{\pi}\frac{\|\boldsymbol{w}^*\|}{\|\boldsymbol{w}\|}\right)\boldsymbol{w} - \frac{k}{\pi}(\pi-\theta)\boldsymbol{w}^*\right]. \quad (3.16)$$

Lemma 5. *(Lipschitz gradient with co-planar assumption, [7])*

Assume $\|\boldsymbol{w}_1\|, \|\boldsymbol{w}_2\| \geq M$, $\boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{w}^*$ are on the same two dimensional half-plane defined by $\boldsymbol{w}^*$, then

$$\|\nabla f(\boldsymbol{w}_1) - \nabla f(\boldsymbol{w}_2)\| \leq L\|\boldsymbol{w}_1 - w_2\|$$

for $L = 1 + \frac{3\|\boldsymbol{w}^*\|}{M}$.

Lemma 6. *For $k \geq 1$, there exists constants $M_k, T > 0$ such that for all $t \geq T$, the iterations of Algorithm 1 satisfy:*

$$\|\boldsymbol{w}^t\| \in [\|\boldsymbol{w}^*\| - M_k, \|\boldsymbol{w}^*\| + M_k]. \quad (3.17)$$

where $M_k < \|\boldsymbol{w}^*\|$, and $M_k \to 0$ as $k \to \infty$.

From Lemma 6, WLOG, we will assume that $T = 0$.

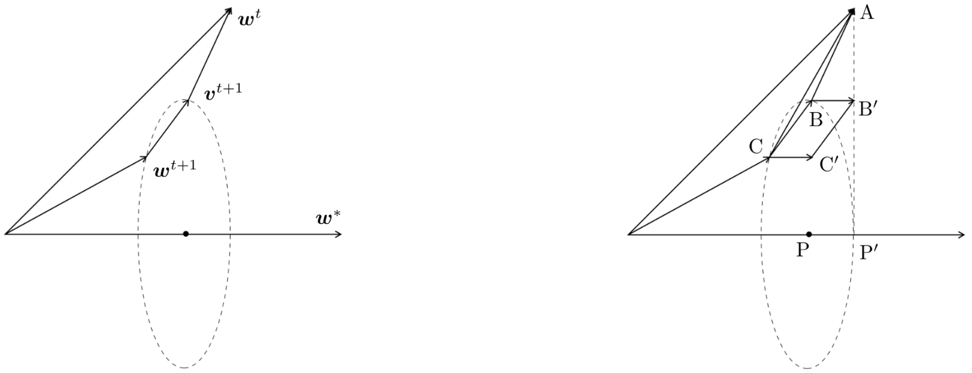


Figure 3.1: Geometry of the update of $\boldsymbol{w}^t$ and the corresponding $\boldsymbol{w}^{t+1}, \boldsymbol{v}^{t+1}$.

Lemma 7. *(Descent of $\mathcal{L}_\beta$ due to $\boldsymbol{w}$ update)*
For η small such that $\eta \leq \frac{1}{\beta+L}$, we have

$$\mathcal{L}_\beta(\boldsymbol{u}^{t+1}, \boldsymbol{w}^{t+1}) \leq \mathcal{L}_\beta(\boldsymbol{w}^t, \boldsymbol{u}^t).$$

3.3.1 Proof of Lemma 3

By Algorithm 1 and Lemma 6, $\|\boldsymbol{w}^t\| \geq \|\boldsymbol{w}^*\| - M > 0$, for all t, and $\boldsymbol{w}^{t+1}$ is in some closed neighborhood of $\boldsymbol{w}^t$. We consider the subspace spanned by $\boldsymbol{w}^t, \boldsymbol{w}^{t+1}$, and $\boldsymbol{w}^*$, this reduces the problem to a 3-dimensional space.
Consider the plane formed by $\boldsymbol{w}^t$ and $\boldsymbol{w}^*$. Let $\boldsymbol{v}^{t+1}$ be the point on this plane, closest to $\boldsymbol{w}^t$, such that $\|\boldsymbol{w}^{t+1}\| = \|\boldsymbol{v}^{t+1}\|$ and $\theta(\boldsymbol{w}^{t+1}, \boldsymbol{w}^*) = \theta(\boldsymbol{v}^{t+1}, \boldsymbol{w}^*)$. In other words, $\boldsymbol{v}^{t+1}$ is the intersection of the plane formed by $\boldsymbol{w}^t, \boldsymbol{w}^*$ and the cone with tip at zero, side length $\|\boldsymbol{w}^{t+1}\|$, and main axis $\boldsymbol{w}^*$ (See Figure 3.1). Then

$$\begin{aligned}\|\nabla f(\boldsymbol{w}^t) - \nabla f(\boldsymbol{w}^{t+1})\| \leq& \|\nabla f(\boldsymbol{w}^t) - \nabla f(\boldsymbol{v}^{t+1})\| + \|\nabla f(\boldsymbol{v}^{t+1}) - \nabla f(\boldsymbol{w}^{t+1})\| \\ \leq& L_1\|\boldsymbol{w}^t - \boldsymbol{v}^{t+1}\| + L_2\|\boldsymbol{v}^{t+1} - \boldsymbol{w}^{t+1}\| \qquad (3.18)\end{aligned}$$

for some constants L_1, L_2. The first term is bounded since $\boldsymbol{w}^t, \boldsymbol{v}^{t+1}, \boldsymbol{w}^*$ are co-planar by construction, and Lemma 5 applies. The second term is bounded by applying Equation 3.16 with $\|\boldsymbol{w}^{t+1}\| = \|\boldsymbol{v}^{t+1}\|$ and $\theta(\boldsymbol{w}^{t+1}, \boldsymbol{w}^*) = \theta(\boldsymbol{v}^{t+1}, \boldsymbol{w}^*)$. It remains to show there exists a constant $L_3 > 0$ such that

$$\|\boldsymbol{w}^t - \boldsymbol{v}^{t+1}\| + \|\boldsymbol{v}^{t+1} - \boldsymbol{w}^{t+1}\| \leq L_3 \|\boldsymbol{w}^t - \boldsymbol{w}^{t+1}\|$$

Let A, B, C be the tips of $\boldsymbol{w}^t, \boldsymbol{v}^{t+1}, \boldsymbol{w}^{t+1}$, respectively. Let P be the point on $\boldsymbol{w}^*$ that is at the base of the cone (so P is the center of the circle with B, C on the arc). We will show there exists a constant L_3 such that

$$|AB| + |BC| \leq L_3 |AC| \tag{3.19}$$

Case 1: A, B, P are collinear: By looking at the cross-section of the plane formed by AB, AC, it can be seen that AC is not the smallest edge in $\triangle ABC$. Thus there exists some L_3 such that Equation 3.19 holds.

Case 2: A, B, P are not collinear: Translate B, C, P to B', C', P' such that A, B', P' are collinear and $BB', CC', PP' /\!/ \boldsymbol{w}^*$. Then by Case 1:

$$|AB'| + |B'C'| \leq L_3 |AC'|$$

and AC' is not the smallest edge in $\triangle AB'C'$. By back-translating B', C' to B, C, it can be seen that AC is again not the smallest edge in $\triangle ABC$. This implies

$$|AB| + |BC| \leq L_4 |AC|$$

for some constant L_4. Thus Equation 3.19 is proved. Combining with Equation 3.18, Lemma 3 is proved. □

3.3.2 Proof of Lemma 6

First we show that if $\|\boldsymbol{w}^t\| < \|\boldsymbol{w}^*\|$, the update of Algorithm 1 will satisfy $\|\boldsymbol{w}^{t+1}\| > \|\boldsymbol{w}^t\|$. By Lemma 4,

$$\begin{aligned}\nabla f(\boldsymbol{w}) =& \frac{1}{k^2}\left[\left(k + \frac{k^2-k}{\pi} - \frac{k}{\pi}\frac{\|\boldsymbol{w}^*\|}{\|\boldsymbol{w}\|}\sin\theta - \frac{k^2-k}{\pi}\frac{\|\boldsymbol{w}^*\|}{\|\boldsymbol{w}\|}\right)\boldsymbol{w} - \frac{k}{\pi}(\pi-\theta)\boldsymbol{w}^*\right] \\ =& \frac{1}{k^2}(C_1\boldsymbol{w} - C_2\boldsymbol{w}^*)\end{aligned}$$

so the update of $\boldsymbol{w}^t$ reads

$$\boldsymbol{w}^{t+1} = \boldsymbol{w}^t - \eta\frac{C_1^t + \beta k^2}{k^2}\boldsymbol{w}^t + \eta\frac{C_2^t}{k^2}\boldsymbol{w}^* + \eta\beta\boldsymbol{u}^{t+1},$$

where $C_2^t > 0$. Since $\boldsymbol{u}^{t+1} = S_{\lambda/\beta}(\boldsymbol{w}^t)$, the term $\eta\beta\boldsymbol{u}^{t+1}$ will increase the norm of $\boldsymbol{w}^t$. For the remaining terms,

$$C_1^t = k + \frac{k^2-k}{\pi} - \frac{k}{\pi}\frac{\|\boldsymbol{w}^*\|}{\|\boldsymbol{w}^t\|}\sin\theta - \frac{k^2-k}{\pi}\frac{\|\boldsymbol{w}^*\|}{\|\boldsymbol{w}^t\|} \leq k + \frac{k^2-k}{\pi}\left(1 - \frac{\|\boldsymbol{w}^*\|}{\|\boldsymbol{w}^t\|}\right)$$

When $\frac{\|\boldsymbol{w}^*\|}{\|\boldsymbol{w}^t\|}$ is large, C_1^t is negative. The update will increase the norm of $\|\boldsymbol{w}^t\|$ if $C_1^t + \beta k^2 \leq 0$ and

$$\left\|\frac{C_1^t + \beta k^2}{k^2}\boldsymbol{w}^t\right\| > \left\|\frac{C_2^t}{k^2}\boldsymbol{w}^*\right\|$$

This condition is satisfied when

$$-\left[k + \frac{k^2-k}{\pi}\left(1 - \frac{\|\boldsymbol{w}^*\|}{\|\boldsymbol{w}^t\|}\right) + \beta k^2\right] > \frac{k}{\pi}\frac{\|\boldsymbol{w}^*\|}{\|\boldsymbol{w}^t\|}$$

When $\frac{\|\boldsymbol{w}^*\|}{\|\boldsymbol{w}^t\|} > 1$, the LHS is $O(k^2)$, while the RHS is $O(k)$. Thus there exists some M_k such that $\boldsymbol{w}^t$ will eventually stay in the region $\|\boldsymbol{w}^t\| \geq \|\boldsymbol{w}^*\| - M_k$. Moreover, as $k \to \infty$, we have $M_k \to 0$.

Next, when $\|\boldsymbol{w}^t\| > \|\boldsymbol{w}^*\|$, the update of $\boldsymbol{w}^t$ reads

$$\boldsymbol{w}^{t+1} = \boldsymbol{w}^t - \eta\frac{C_1^t}{k^2}\boldsymbol{w}^t + \eta\frac{C_2^t}{k^2}\boldsymbol{w}^* - \eta\beta(\boldsymbol{w}^t - \boldsymbol{u}^{t+1})$$

the last term decreases the norm of $\boldsymbol{w}^t$. In this case, C_1^t is positive and

$$C_1^t \geq \frac{k\pi - k}{\pi} + \frac{k^2 - k}{\pi}\left(1 - \frac{\|\boldsymbol{w}^*\|}{\|\boldsymbol{w}^t\|}\right)$$

The update will decrease the norm of $\boldsymbol{w}^t$ if

$$\frac{k\pi - k}{\pi} + \frac{k^2 - k}{\pi}\left(1 - \frac{\|\boldsymbol{w}^*\|}{\|\boldsymbol{w}^t\|}\right) > \frac{k}{\pi}\frac{\|\boldsymbol{w}^*\|}{\|\boldsymbol{w}^t\|}$$

which holds when $\frac{\|\boldsymbol{w}^*\|}{\|\boldsymbol{w}^t\|} < 1$, and the Lemma is proved. □

3.3.3 Proof of Lemma 7

By the update of $\boldsymbol{u}^t$, $\mathcal{L}_\beta(\boldsymbol{w}^t, \boldsymbol{u}^{t+1}) \leq \mathcal{L}_\beta(\boldsymbol{w}^t, \boldsymbol{u}^t)$. For the update of $\boldsymbol{w}^t$, notice that

$$\nabla f(\boldsymbol{w}^t) = \frac{1}{\eta}\left(\boldsymbol{w}^t - \boldsymbol{w}^{t+1}\right) - \beta(\boldsymbol{w}^t - \boldsymbol{u}^{t+1})$$

Then for a fixed $\boldsymbol{u} := \boldsymbol{u}^{t+1}$, we have

$$
\begin{aligned}
& \mathcal{L}_\beta(\boldsymbol{w}^{t+1}, \boldsymbol{u}) - \mathcal{L}_\beta(\boldsymbol{w}^t, \boldsymbol{u}) \\
=& f(\boldsymbol{w}^{t+1}) - f(\boldsymbol{w}^t) + \frac{\beta}{2}\left(\|\boldsymbol{w}^{t+1} - \boldsymbol{u}\|^2 - \|\boldsymbol{w}^t - \boldsymbol{u}\|^2\right) \\
\leq& \langle \nabla f(\boldsymbol{w}^t), \boldsymbol{w}^{t+1} - \boldsymbol{w}^t\rangle + \frac{L}{2}\|\boldsymbol{w}^{t+1} - \boldsymbol{w}^t\|^2 + \frac{\beta}{2}\left(\|\boldsymbol{w}^{t+1} - \boldsymbol{u}\|^2 - \|\boldsymbol{w}^t - \boldsymbol{u}\|^2\right) \\
=& \frac{1}{\eta}\langle \boldsymbol{w}^t - \boldsymbol{w}^{t+1}, \boldsymbol{w}^{t+1} - \boldsymbol{w}^t\rangle - \beta\langle \boldsymbol{w}^t - \boldsymbol{u}, \boldsymbol{w}^{t+1} - \boldsymbol{w}^t\rangle + \frac{L}{2}\|\boldsymbol{w}^{t+1} - \boldsymbol{w}^t\|^2 \\
+& \frac{\beta}{2}\left(\|\boldsymbol{w}^{t+1} - \boldsymbol{u}\|^2 - \|\boldsymbol{w}^t - \boldsymbol{u}\|^2\right) \\
=& \frac{1}{\eta}\langle \boldsymbol{w}^t - \boldsymbol{w}^{t+1}, \boldsymbol{w}^{t+1} - \boldsymbol{w}^t\rangle + \left(\frac{L}{2} + \frac{\beta}{2}\right)\|\boldsymbol{w}^{t+1} - \boldsymbol{w}^t\|^2 + \frac{\beta}{2}\|\boldsymbol{w}^{t+1} - \boldsymbol{u}\|^2 - \frac{\beta}{2}\|\boldsymbol{w}^t - \boldsymbol{u}\|^2 \\
-& \beta\langle \boldsymbol{w}^t - \boldsymbol{u}, \boldsymbol{w}^{t+1} - \boldsymbol{w}^t\rangle - \frac{\beta}{2}\|\boldsymbol{w}^{t+1} - \boldsymbol{w}^t\|^2 \\
=& \left(\frac{L}{2} + \frac{\beta}{2} - \frac{1}{\eta}\right)\|\boldsymbol{w}^{t+1} - \boldsymbol{w}^t\|^2
\end{aligned}
$$

Therefore, when η is small such that $\eta \leq \frac{2}{\beta + L}$, the update on $\boldsymbol{w}^t$ will decrease $\mathcal{L}_\beta$. □

3.3.4 Proof of Theorem 3.1

We will first show that if $\theta(\boldsymbol{w}^0, \boldsymbol{w}^*) \leq \pi - \delta$, then $\theta(\boldsymbol{w}^t, \boldsymbol{w}^*) \leq \pi - \delta$, for all t. We will show $\theta(\boldsymbol{w}^1, \boldsymbol{w}^*) \leq \pi - \delta$, the statement is then followed by induction. To this end, by the update of $\boldsymbol{w}^t$, one has

$$
\begin{aligned}
\boldsymbol{w}^1 &= C\boldsymbol{w}^0 + \left(\eta\frac{\pi - \theta(\boldsymbol{w}^0, \boldsymbol{w}^*)}{k\pi}\right)\boldsymbol{w}^* + \eta\beta\boldsymbol{u}^1 \\
&= C\boldsymbol{w}^0 + \eta\frac{\pi - \theta(\boldsymbol{w}^0, \boldsymbol{w}^*)}{k\pi}\boldsymbol{w}^* + \eta\beta\boldsymbol{u}^1
\end{aligned}
$$

for some constant $C > 0$. Since $\boldsymbol{u}^1 = S_{\lambda/\beta}(\boldsymbol{w}^0), \theta(\boldsymbol{u}^1, \boldsymbol{w}^0) \leq \frac{\pi}{2}$. Notice that the sum of the first two terms on the RHS brings the vector closer to $\boldsymbol{w}^*$, while the last term may behave unexpectedly. Consider the worst case scenario: $\boldsymbol{w}^0, \boldsymbol{w}^*, \boldsymbol{u}^1$ are co-planar with $\theta(\boldsymbol{u}^1, \boldsymbol{w}^0) = \frac{\pi}{2}$,

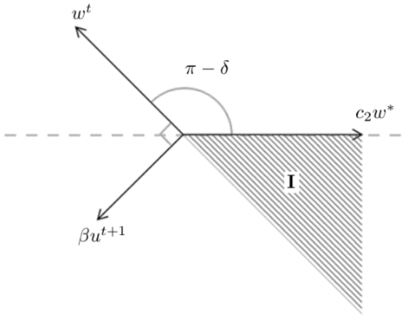


Figure 3.2: Worst case of the update on $\boldsymbol{w}^t$

and $\boldsymbol{w}^*, \boldsymbol{u}^1$ are on two sides of $\boldsymbol{w}^0$ (See Figure 3.2). We need $\frac{\delta}{k\pi}\boldsymbol{w}^* + \beta \boldsymbol{u}^1$ to be in region I. This condition is satisfied when β is small such that

$$\sin\delta \geq \frac{\beta \|\boldsymbol{u}^1\|}{\frac{\delta}{k\pi}\|\boldsymbol{w}^*\|} = \frac{k\pi\beta\|\boldsymbol{u}^1\|}{\delta}$$

since $\|\boldsymbol{u}^1\| \leq 1$, it is sufficient to have $\beta \leq \frac{\delta \sin\delta}{k\pi}$.

Next, consider the limit of the RVSM algorithm. Since $\mathcal{L}_\beta(\boldsymbol{w}^t, \boldsymbol{u}^t)$ is non-negative, by Lemma 7, $\mathcal{L}_\beta$ converges to some limit $\mathcal{L}$. This implies $(\boldsymbol{w}^t, \boldsymbol{u}^t)$ converges to some stationary point $(\bar{\boldsymbol{w}}, \bar{\boldsymbol{u}})$. By Lemma 4 and the update of $\boldsymbol{w}^t$, we have

$$\overline{\boldsymbol{w}} = c_1\overline{\boldsymbol{w}} + \eta c_2 \boldsymbol{w}^* + \eta\beta\overline{\boldsymbol{u}} \tag{3.20}$$

for some constant $c_1 > 0, c_2 \geq 0$, where $c_2 = \frac{\pi - \theta}{k\pi}$, with $\theta := \theta(\bar{\boldsymbol{w}}, \boldsymbol{w}^*)$, and $\bar{\boldsymbol{u}} = S_{\lambda/\beta}(\bar{\boldsymbol{w}})$. If $c_2 = 0$, then we must have $\bar{\boldsymbol{w}} /\!/ \bar{\boldsymbol{u}}$. But since $\bar{\boldsymbol{u}} = S_{\lambda/\beta}$, this implies all non-zero components of $\bar{\boldsymbol{w}}$ are either equal in magnitude, or all have magnitude smaller than $\frac{\lambda}{\beta}$. The latter case is not possible when $\frac{\lambda}{\beta} < \frac{1}{\sqrt{d}}$. Furthermore, $c_2 = 0$ when $\theta(\bar{\boldsymbol{w}}, \boldsymbol{w}^*) = \pi$ or 0. We have shown that $\theta(\bar{\boldsymbol{w}}, \boldsymbol{w}^*) \leq \pi - \delta$, thus $\theta(\bar{\boldsymbol{w}}, \boldsymbol{w}^*) = 0$. Thus, $\bar{\boldsymbol{w}} = \boldsymbol{w}^*$, and all non-zero components of $\boldsymbol{w}^*$ are equal in magnitude. This has probability zero if we assume $\boldsymbol{w}^*$ is initiated uniformly on the unit circle. Hence we will assume that almost surely, $c_2 > 0$. Expression (3.20) implies

$$(c_2\boldsymbol{w}^* + \beta\bar{\boldsymbol{u}}) /\!/ \bar{\boldsymbol{w}} \tag{3.21}$$

Expression (3.21) implies $\bar{\boldsymbol{w}}, \bar{\boldsymbol{u}}$, and $\boldsymbol{w}^*$ are co-planar. Let $\gamma := \theta(\bar{\boldsymbol{w}}, \bar{\boldsymbol{u}})$. From expression (3.21), and the assumption that $\|\boldsymbol{w}^*\| = 1$, we have

$$(\langle c_2\boldsymbol{w}^* + \beta\bar{\boldsymbol{u}}, \bar{\boldsymbol{w}}\rangle)^2 = \|c_2\boldsymbol{w}^* + \beta\bar{\boldsymbol{u}}\|^2\|\bar{\boldsymbol{w}}\|^2$$

or

$$\begin{aligned} &\|\bar{\boldsymbol{w}}\|^2(c_2^2\cos^2\theta + 2c_2\beta\|\bar{\boldsymbol{u}}\|\cos\theta\cos\gamma + \beta^2\|\bar{\boldsymbol{u}}\|^2\cos^2\gamma) \\ =&\|\bar{\boldsymbol{w}}\|^2(c_2^2 + 2c_2\beta\|\bar{\boldsymbol{u}}\|\cos(\theta+\gamma) + \beta^2\|\bar{\boldsymbol{u}}\|^2) \end{aligned}$$

This reduces to

$$c_2^2\sin^2\theta - 2c_2\beta\|\bar{\boldsymbol{u}}\|\sin\theta\sin\gamma + \beta^2\|\bar{\boldsymbol{u}}\|^2\sin^2\gamma = 0,$$

which implies $\frac{\pi-\theta}{k\pi}\sin\theta = \beta\|\bar{\boldsymbol{u}}\|\sin\gamma$. By the initialization $\beta \le \frac{\delta\sin\delta}{k\pi}$, we have $\frac{\pi-\theta}{k\pi}\sin\theta < \frac{\delta}{k\pi}\sin\delta$. This implies $\theta < \delta$.

Finally, the limit point satisfies $\|\nabla f(\bar{\boldsymbol{w}}) + \beta(\bar{\boldsymbol{w}} - \bar{\boldsymbol{u}})\| = 0$. By the initialization requirement, we have $\|\beta(\bar{\boldsymbol{w}} - \bar{\boldsymbol{u}})\| < \beta \le \frac{\delta\sin\delta}{k\pi}$. This implies $\|\nabla f(\bar{\boldsymbol{w}})\| \le \frac{\delta\sin\delta}{k\pi}$. By the Lipschitz gardient property in Lemma 3 and critical points property in Lemma 4, $\bar{\boldsymbol{w}}$ must be close to $\boldsymbol{w}^*$. In other words, $\|\bar{\boldsymbol{w}} - \boldsymbol{w}^*\|$ is comparable to the chord length of the circle of radius $\|\boldsymbol{w}^*\|$ and angle θ:

$$\|\bar{\boldsymbol{w}} - \boldsymbol{w}^*\| = O\left(2\sin\left(\frac{\theta}{2}\right)\right) = O(\sin\theta) = O\left(\frac{k\pi\beta\|\bar{\boldsymbol{u}}\|\sin\gamma}{\pi-\theta}\right) = O(k\beta\sin\gamma). \qquad \square$$

3.4 Proof of the Second Convergence Result

The following Lemmas give an outline for the proof of Theorem 3.2.

Lemma 8. *If every entry of $\boldsymbol{Z}$ is i.i.d. sampled from $\mathcal{N}(0,1)$, $\|\boldsymbol{w}^*\| = 1$, and $\|\boldsymbol{w}\| \neq 0$, then the true gradient of the population loss $f(\boldsymbol{w})$ is*

$$\nabla f(\boldsymbol{w}) = \frac{-k}{2\pi\|\boldsymbol{w}\|} \frac{\left(\boldsymbol{I} - \frac{\boldsymbol{w}\boldsymbol{w}^T}{\|\boldsymbol{w}\|^2}\right)\boldsymbol{w}^*}{\left\|\left(\boldsymbol{I} - \frac{\boldsymbol{w}\boldsymbol{w}^T}{\|\boldsymbol{w}\|^2}\right)\boldsymbol{w}^*\right\|}, \tag{3.22}$$

for $\theta(\boldsymbol{w}, \boldsymbol{w}^*) \in (0, \pi)$; and the expected coarse gradient w.r.t. $\boldsymbol{w}$ is

$$\mathbb{E}_{\boldsymbol{Z}}[g(\boldsymbol{w}, \boldsymbol{Z})] = \frac{k}{\pi}\left[\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|} - \cos\left(\frac{\theta(\boldsymbol{w}, \boldsymbol{w}^*)}{2}\right)\frac{\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|} + \boldsymbol{w}^*}{\left\|\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|} + \boldsymbol{w}^*\right\|}\right] \tag{3.23}$$

Lemma 9. *(Properties of true gradient)*
Given $\boldsymbol{w}_1, \boldsymbol{w}_2$ with $\min\{\|\boldsymbol{w}_1\|, \|\boldsymbol{w}_2\|\} = c > 0$ and $\max\{\|\boldsymbol{w}_1\|, \|\boldsymbol{w}_2\|\} = C$, there exists a constant $L_f > 0$ depends on c and C such that

$$\|\nabla f(\boldsymbol{w}_1) - \nabla f(\boldsymbol{w}_2)\| \leq L_f \|\boldsymbol{w}_1 - \boldsymbol{w}_2\|$$

Moreover, we have

$$f(\boldsymbol{w}_2) \leq f(\boldsymbol{w}_1) + \langle \nabla f(\boldsymbol{w}_1), \boldsymbol{w}_2 - \boldsymbol{w}_1 \rangle + \frac{L_f}{2}\|\boldsymbol{w}_2 - \boldsymbol{w}_1\|^2.$$

Lemma 10. *(Properties of expected coarse gradient)*
If $\boldsymbol{w}_1, \boldsymbol{w}_2$ satisfy $\frac{1}{2} \leq \|\boldsymbol{w}_1\|, \|\boldsymbol{w}_2\| \leq \frac{3}{2}$, and $\theta(\boldsymbol{w}_1, \boldsymbol{w}^*), \theta(\boldsymbol{w}_2, \boldsymbol{w}^*) \in (0, \pi)$, then there exists a constant K such that

$$\|\mathbb{E}_{\boldsymbol{Z}}[g(\boldsymbol{w}_1, \boldsymbol{Z})] - \mathbb{E}_{\boldsymbol{Z}}[g(\boldsymbol{w}_2, \boldsymbol{Z})]\| \leq K\|\boldsymbol{w}_1 - \boldsymbol{w}_2\| \tag{3.24}$$

Moreover, there exists a constant L such that

$$f(\boldsymbol{w}_2) - f(\boldsymbol{w}_1) \leq \langle \mathbb{E}_{\boldsymbol{Z}}[g(\boldsymbol{w}_1, \boldsymbol{Z})], \boldsymbol{w}_2 - \boldsymbol{w}_1 \rangle + \frac{L}{2}\|\boldsymbol{w}_2 - \boldsymbol{w}_1\|^2. \tag{3.25}$$

Remark. *The condition $\frac{1}{2} \leq \|\boldsymbol{w}_1\|, \|\boldsymbol{w}_2\| \leq \frac{3}{2}$ in Lemma 10 is to match the RVSCGD algorithm and to give an explicit value for K. The result still holds in general when $0 < c \leq \|\boldsymbol{w}_1\|, \|\boldsymbol{w}_2\| \leq C$. Compared to Lemma 9, when $c = \frac{1}{2}$ and $C = \frac{3}{2}$, one has $L_f = \frac{4\sqrt{k}}{\pi}$, which is a sharper bound than $K = \frac{k}{\sqrt{2\pi}}$ in the coarse gradient case.*

Lemma 11. *(Angle Descent)*
Let $\theta^t := \theta(\boldsymbol{w}^t, \boldsymbol{w}^*)$. Suppose the initialization of the RVSCGD algorithm satisfies $\theta^0 \leq \pi - \delta$ and $\beta \leq \frac{k \sin \delta}{2\sqrt{2\pi}}$, then $\theta^{t+1} \leq \theta^t$.

Lemma 12. *(Lagrangian Descent)*
Suppose the initialization of the RVSCGD algorithm satisfies $\eta \leq \frac{1}{\beta + L}$, where L is the Lipschitz constant in Lemma 10, then $\mathcal{L}_\beta(\boldsymbol{u}^{t+1}, \boldsymbol{w}^{t+1}) \leq \mathcal{L}_\beta(\boldsymbol{u}^t, \boldsymbol{w}^t)$.

Lemma 13. *(Properties of limit point)*
Suppose the initialization of the RVSCGD algorithm satisfies: $\theta(\boldsymbol{w}^0, \boldsymbol{w}^*) \leq \pi - \delta$, for some $\delta > 0$, λ is small such that $\frac{2\sqrt{2\pi}}{k}\lambda\sqrt{d} < 1$, and η is small such that $\eta\frac{k}{2\sqrt{2\pi}} < 1$. Let $\theta := \theta(\bar{\boldsymbol{w}}, \boldsymbol{w}^*)$ and $\gamma := \theta(\bar{\boldsymbol{u}}, \bar{\boldsymbol{w}})$, then $(\boldsymbol{u}^t, \boldsymbol{w}^t)$ converges to a limit point $(\bar{\boldsymbol{u}}, \bar{\boldsymbol{w}})$ such that

$$\theta < \delta \qquad \text{and} \qquad \|\boldsymbol{w}^* - \bar{\boldsymbol{w}}\| \leq \frac{4\sqrt{2\pi}\beta \sin \gamma}{k}.$$

Lemmas 8, 9 follow directly from [73]. The proof of Lemmas 10, 11, 12, 13 are provided below.

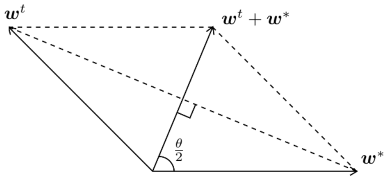


Figure 3.3: Geometry of $\boldsymbol{w}^t$ and $\boldsymbol{w}^*$ when $\|\boldsymbol{w}^t\| = \|\boldsymbol{w}^*\| = 1$.

3.4.1 Proof of Lemma 10

First suppose $\|\boldsymbol{w}_1\| = \|\boldsymbol{w}_2\| = 1$. By Lemma 5.3 of [73], we have

$$\mathbb{E}_{\boldsymbol{Z}}[g(\boldsymbol{w}_j, \boldsymbol{Z})] = \frac{k}{\pi}\left[\boldsymbol{w}_j - \cos\left(\frac{\theta(\boldsymbol{w}_j, \boldsymbol{w}^*)}{2}\right)\frac{\boldsymbol{w}_j + \boldsymbol{w}^*}{\|\boldsymbol{w}_j + \boldsymbol{w}^*\|}\right]$$

for $j = 1, 2$. Consider the plane formed by $\boldsymbol{w}_j$ and $\boldsymbol{w}^*$, since $\|\boldsymbol{w}^*\| = 1$, we have an equilateral triangle formed by $\boldsymbol{w}_j$ and $\boldsymbol{w}^*$ (See Fig. 3.3).

Simple geometry shows

$$\cos\left(\frac{\theta(\boldsymbol{w}_j, \boldsymbol{w}^*)}{2}\right) = \frac{\frac{1}{2}\|\boldsymbol{w}_j + \boldsymbol{w}^*\|}{\|\boldsymbol{w}^*\|} = \frac{1}{2}\|\boldsymbol{w}_j + \boldsymbol{w}^*\|$$

Thus the expected coarse gradient simplifies to

$$\mathbb{E}_{\boldsymbol{Z}}[g(\boldsymbol{w}_j, \boldsymbol{Z})] = \frac{k}{\pi}\left[\boldsymbol{w}_j - \frac{\boldsymbol{w}_j + \boldsymbol{w}^*}{2}\right] = \frac{k}{2\pi}\boldsymbol{w}_j - \frac{k}{2\pi}\boldsymbol{w}^* \tag{3.26}$$

which implies

$$\|\mathbb{E}_{\boldsymbol{Z}}[g(\boldsymbol{w}_1, \boldsymbol{Z})] - \mathbb{E}_{\boldsymbol{Z}}[g(\boldsymbol{w}_2, \boldsymbol{Z})]\| \leq K\|\boldsymbol{w}_1 - \boldsymbol{w}_2\| \tag{3.27}$$

with $K = \frac{k}{2\pi}$.

Now suppose $\frac{1}{2} \leq \|\boldsymbol{w}_1\|, \|\boldsymbol{w}_2\| \leq \frac{3}{2}$. By equation (3.23), we have $\mathbb{E}_{\boldsymbol{Z}}[g(\boldsymbol{w}, \boldsymbol{Z})] = \mathbb{E}_{\boldsymbol{Z}}[g(\frac{\boldsymbol{w}}{C}, \boldsymbol{Z})]$, for all $C > 0$. Then

$$\begin{aligned}\|\mathbb{E}_{\boldsymbol{Z}}[g(\boldsymbol{w}_1, \boldsymbol{Z})] - \mathbb{E}_{\boldsymbol{Z}}[g(\boldsymbol{w}_2, \boldsymbol{Z})]\| =& \left\|\mathbb{E}_{\boldsymbol{Z}}\left[g\left(\frac{\boldsymbol{w}_1}{\|\boldsymbol{w}_1\|}, \boldsymbol{Z}\right)\right] - \mathbb{E}_{\boldsymbol{Z}}\left[g\left(\frac{\boldsymbol{w}_2}{\|\boldsymbol{w}_2\|}, \boldsymbol{Z}\right)\right]\right\| \\ \leq& K'\left\|\frac{\boldsymbol{w}_1}{\|\boldsymbol{w}_1\|} - \frac{\boldsymbol{w}_2}{\|\boldsymbol{w}_2\|}\right\| \\ \leq& 2K'\|\boldsymbol{w}_1 - \boldsymbol{w}_2\|\end{aligned}$$

where the first inequality follows from (3.27), and the second inequality is from the constraint $\frac{1}{2} \leq \|\boldsymbol{w}_1\|, \|\boldsymbol{w}_2\| \leq \frac{3}{2}$, with equality when $\|\boldsymbol{w}_1\| = \|\boldsymbol{w}_2\| = \frac{1}{2}$. Letting $K = 2K' = \frac{k}{\pi}$, the first claim is proved.

It remains to show the gradient descent inequality. By [73], we have

$$f(\boldsymbol{w}) = \frac{1}{8}\left[\mathbf{1}^T(I + \mathbf{1}\mathbf{1}^T)\mathbf{1} - 2\mathbf{1}^T\left(\left(1 - \frac{2}{\pi}\theta(\boldsymbol{w}, \boldsymbol{w}^*)\right)I + \mathbf{1}\mathbf{1}^T\right)\mathbf{1} + \mathbf{1}^T(I + \mathbf{1}\mathbf{1}^T)\mathbf{1}\right]$$

Let $\theta_1 = \theta(\boldsymbol{w}_1, \boldsymbol{w}^*), \theta_2 = \theta(\boldsymbol{w}_2, \boldsymbol{w}^*)$. Then

$$f(\boldsymbol{w}_2) - f(\boldsymbol{w}_1) = \frac{1}{4}\left[\mathbf{1}^T\left(\left(\frac{2}{\pi}\theta_2 - \frac{2}{\pi}\theta_1\right)I\right)\mathbf{1}\right] = \frac{k}{2\pi}(\theta_2 - \theta_1)$$

We will show

$$f(\boldsymbol{w}_2) - f(\boldsymbol{w}_1) \leq \langle\mathbb{E}_{\boldsymbol{Z}}[g(\boldsymbol{w}_1, \boldsymbol{Z})], \boldsymbol{w}_2 - \boldsymbol{w}_1\rangle + L\|\boldsymbol{w}_2 - \boldsymbol{w}_1\|^2$$

for $\|\boldsymbol{w}_1\| = \|\boldsymbol{w}_2\| = 1$ and $\theta_2 \leq \theta_1$. By equation (3.26),

$$\mathbb{E}_{\boldsymbol{Z}}[g(\boldsymbol{w}_1, \boldsymbol{Z})] = \frac{k}{2\pi}\left(\boldsymbol{w}_1 - \boldsymbol{w}^*\right)$$

It remains to show

$$\frac{k}{2\pi}(\theta_2 - \theta_1) \leq \left\langle \frac{k}{2\pi}\left(\boldsymbol{w}_1 - \boldsymbol{w}^*\right), \boldsymbol{w}_2 - \boldsymbol{w}_1 \right\rangle + L\|\boldsymbol{w}_2 - \boldsymbol{w}_1\|^2$$

or there exists a constant K_1 such that

$$\theta_2 - \theta_1 \leq \langle \boldsymbol{w}_1 - \boldsymbol{w}^*, \boldsymbol{w}_2 - \boldsymbol{w}_1 \rangle + K_1\|\boldsymbol{w}_2 - \boldsymbol{w}_1\|^2$$

Notice that by writing $K_1 = \frac{1}{2} + K_2$, we have

$$\begin{aligned}
&\langle \boldsymbol{w}_1 - \boldsymbol{w}^*, \boldsymbol{w}_2 - \boldsymbol{w}_1 \rangle + K_1\|\boldsymbol{w}_2 - \boldsymbol{w}_1\|^2 \\
=&\langle \boldsymbol{w}_1 - \boldsymbol{w}^*, \boldsymbol{w}_2 - \boldsymbol{w}_1 \rangle + K_1\langle \boldsymbol{w}_2 - \boldsymbol{w}_1, \boldsymbol{w}_2 - \boldsymbol{w}_1 \rangle \\
=&\langle \boldsymbol{w}_1 - \boldsymbol{w}^*, \boldsymbol{w}_2 - \boldsymbol{w}_1 \rangle + \frac{1}{2}\langle \boldsymbol{w}_2 - \boldsymbol{w}_1, \boldsymbol{w}_2 - \boldsymbol{w}_1 \rangle + K_2\|\boldsymbol{w}_2 - \boldsymbol{w}_1\|^2 \\
=&\langle \frac{1}{2}\boldsymbol{w}_1 + \frac{1}{2}\boldsymbol{w}_2 - \boldsymbol{w}^*, \boldsymbol{w}_2 - \boldsymbol{w}_1 \rangle + K_2\|\boldsymbol{w}_2 - \boldsymbol{w}_1\|^2 \\
=&\langle -\boldsymbol{w}^*, \boldsymbol{w}_2 - \boldsymbol{w}_1 \rangle + \frac{1}{2}\langle \boldsymbol{w}_1 + \boldsymbol{w}_2, \boldsymbol{w}_2 - \boldsymbol{w}_1 \rangle + K_2\|\boldsymbol{w}_2 - \boldsymbol{w}_1\|^2 \\
=&\langle -\boldsymbol{w}^*, \boldsymbol{w}_2 - \boldsymbol{w}_1 \rangle + K_2\|\boldsymbol{w}_2 - \boldsymbol{w}_1\|^2
\end{aligned}$$

where the last equality follows since $\|\boldsymbol{w}_1\| = \|\boldsymbol{w}_2\| = 1$ implies $\langle \boldsymbol{w}_1 + \boldsymbol{w}_2, \boldsymbol{w}_2 - \boldsymbol{w}_1 \rangle = 0$. On the other hand,

$$\langle -\boldsymbol{w}^*, \boldsymbol{w}_2 - \boldsymbol{w}_1 \rangle = -\|\boldsymbol{w}^*\|\|\boldsymbol{w}_2\|\cos\theta_2 + \|\boldsymbol{w}^*\|\|\boldsymbol{w}_1\|\cos\theta_1 = \cos\theta_1 - \cos\theta_2$$

so it suffices to show there exists a constant K_2 such that

$$\theta_2 + \cos\theta_2 - \theta_1 - \cos\theta_1 \leq K_2\|\boldsymbol{w}_2 - \boldsymbol{w}_1\|^2$$

Notice the function $\theta \mapsto \theta + \cos\theta$ is monotonically increasing on $[0, \pi]$. For $\theta_1, \theta_2 \in [0, \pi]$ with $\theta_2 \leq \theta_1$, the LHS is non-positive, and the inequality holds. Thus one can take $K_2 = 0, K_1 = \frac{1}{2}$, and $L = \frac{k}{4\pi}$. $\square$

3.4.2 Proof of Lemma 11

Due to normalization in the RVSCGD algorithm, $\|\boldsymbol{w}^t\| = 1$ for all t. By equation (3.26), we have

$$\boldsymbol{w}^t - \eta \mathbb{E}_{\boldsymbol{Z}}[g(\boldsymbol{w}^t, \boldsymbol{Z})] = \left(1 - \eta \frac{k}{2\sqrt{2\pi}}\right) \boldsymbol{w}^t + \eta \frac{k}{2\sqrt{2\pi}} \boldsymbol{w}^*$$

and the update of $\boldsymbol{u}$ is the well-known soft-thresholding of $\boldsymbol{w}$ [16, 14]:

$$\boldsymbol{u}^{t+1} = \arg\min_{\boldsymbol{u}} \mathcal{L}_\beta(\boldsymbol{u}, \boldsymbol{w}^t) = S_{\lambda/\beta}(\boldsymbol{w}^t)$$

where $S_{\lambda/\beta}(w) := sgn(w)(|w| - \lambda/\beta)\chi_{\{|w|>\lambda/\beta\}}$ is the soft-thresholding operator; and $S_{\lambda/\beta}(\boldsymbol{w})$ applies the thresholding to each component of $\boldsymbol{w}$. Then the update of $\boldsymbol{w}$ has the form

$$\boldsymbol{w}^{t+1} = C^t \boldsymbol{w}^t + \eta \frac{k}{2\sqrt{2\pi}} \boldsymbol{w}^* + \eta\beta \boldsymbol{u}^{t+1}$$

for some constant $C^t > 0$. Suppose the initialization satisfies $\theta(\boldsymbol{w}^0, \boldsymbol{w}^*) \leq \pi - \delta$, for some $\delta > 0$. It suffices to show that if $\theta^t \leq \pi - \delta$, then $\theta^{t+1} \leq \pi - \delta$. To this end, since $\boldsymbol{u}^{t+1} = S_{\lambda/\beta}(\boldsymbol{w}^t)$, we have $\theta(\boldsymbol{w}^t, \boldsymbol{u}^{t+1}) \leq \frac{\pi}{2}$. Consider the worst case scenario: $\boldsymbol{w}^t, \boldsymbol{w}^*, \boldsymbol{u}^{t+1}$ are co-planar with $\theta(\boldsymbol{u}^{t+1}, \boldsymbol{w}^t) = \frac{\pi}{2}$, and $\boldsymbol{w}^*, \boldsymbol{u}^{t+1}$ are on two sides of $\boldsymbol{w}^t$ (See Figure 3). We need $\frac{k}{2\sqrt{2\pi}} \boldsymbol{w}^* + \beta \boldsymbol{u}^{t+1}$ to be in region I. This condition is satisfied when β is small such that

$$\sin\delta \geq \frac{\beta \|\boldsymbol{u}^{t+1}\|}{\frac{k}{2\sqrt{2\pi}} \|\boldsymbol{w}^*\|} = \frac{2\sqrt{2\pi}\beta \|\boldsymbol{u}^{t+1}\|}{k}$$

or $\beta \leq \frac{k \sin\delta}{2\sqrt{2\pi}\|\boldsymbol{u}^{t+1}\|}$. Since $\boldsymbol{u}^{t+1} = S_{\lambda/\beta}(\boldsymbol{w}^t)$, we have $\|u^{t+1}\| \leq 1$. Thus it suffices to have $\beta \leq \frac{k \sin\delta}{2\sqrt{2\pi}}$. □

3.4.3 Proof of Lemma 12

By definition of the update on $\boldsymbol{u}$, we have $\mathcal{L}_\beta(\boldsymbol{u}^{t+1}, \boldsymbol{w}^t) \leq \mathcal{L}_\beta(\boldsymbol{u}^t, \boldsymbol{w}^t)$. It remains to show $\mathcal{L}_\beta(\boldsymbol{u}^{t+1}, \boldsymbol{w}^{t+1}) \leq \mathcal{L}_\beta(\boldsymbol{u}^{t+1}, \boldsymbol{w}^t)$. First notice that since

$$\boldsymbol{w}^{t+1} = C^t(\boldsymbol{w}^t - \eta\mathbb{E}_{\boldsymbol{Z}}[g(\boldsymbol{w}^t, \boldsymbol{Z})] - \eta\beta(\boldsymbol{w}^t - \boldsymbol{u}^{t+1}))$$

where $C^t > 0$ is the normalizing constant, thus

$$\mathbb{E}_{\boldsymbol{Z}}[g(\boldsymbol{w}^t, \boldsymbol{Z})] = \frac{1}{\eta}\left(\boldsymbol{w}^t - \frac{\boldsymbol{w}^{t+1}}{C^t}\right) - \beta(\boldsymbol{w}^t - \boldsymbol{u}^{t+1})$$

For a fixed $\boldsymbol{u} := \boldsymbol{u}^{t+1}$ we have

$$\begin{aligned}
&\mathcal{L}_\beta(\boldsymbol{u}, \boldsymbol{w}^{t+1}) - \mathcal{L}_\beta(\boldsymbol{u}, \boldsymbol{w}^t) \\
=&f(\boldsymbol{w}^{t+1}) - f(\boldsymbol{w}^t) + \frac{\beta}{2}\left(\|\boldsymbol{w}^{t+1} - \boldsymbol{u}\|^2 - \|\boldsymbol{w}^t - \boldsymbol{u}\|^2\right) \\
\leq&\langle\mathbb{E}_{\boldsymbol{Z}}[g(\boldsymbol{w}^t, \boldsymbol{Z})], \boldsymbol{w}^{t+1} - \boldsymbol{w}^t\rangle + \frac{L}{2}\|\boldsymbol{w}^{t+1} - \boldsymbol{w}^t\|^2 + \frac{\beta}{2}\left(\|\boldsymbol{w}^{t+1} - \boldsymbol{u}\|^2 - \|\boldsymbol{w}^t - \boldsymbol{u}\|^2\right) \\
=&\frac{1}{\eta}\langle\boldsymbol{w}^t - \frac{\boldsymbol{w}^{t+1}}{C^t}, \boldsymbol{w}^{t+1} - \boldsymbol{w}^t\rangle - \beta\langle\boldsymbol{w}^t - \boldsymbol{u}, \boldsymbol{w}^{t+1} - \boldsymbol{w}^t\rangle \\
+&\frac{L}{2}\|\boldsymbol{w}^{t+1} - \boldsymbol{w}^t\|^2 + \frac{\beta}{2}\left(\|\boldsymbol{w}^{t+1} - \boldsymbol{u}\|^2 - \|\boldsymbol{w}^t - \boldsymbol{u}\|^2\right) \\
=&\frac{1}{\eta}\langle\boldsymbol{w}^t - \frac{\boldsymbol{w}^{t+1}}{C^t}, \boldsymbol{w}^{t+1} - \boldsymbol{w}^t\rangle + \left(\frac{L}{2} + \frac{\beta}{2}\right)\|\boldsymbol{w}^{t+1} - \boldsymbol{w}^t\|^2 \\
+&\frac{\beta}{2}\|\boldsymbol{w}^{t+1} - \boldsymbol{u}\|^2 - \frac{\beta}{2}\|\boldsymbol{w}^t - \boldsymbol{u}\|^2 - \beta\langle\boldsymbol{w}^t - \boldsymbol{u}, \boldsymbol{w}^{t+1} - \boldsymbol{w}^t\rangle - \frac{\beta}{2}\|\boldsymbol{w}^{t+1} - \boldsymbol{w}^t\|^2 \\
=&\frac{1}{\eta}\langle\boldsymbol{w}^t - \frac{\boldsymbol{w}^{t+1}}{C^t}, \boldsymbol{w}^{t+1} - \boldsymbol{w}^t\rangle + \left(\frac{L}{2} + \frac{\beta}{2}\right)\|\boldsymbol{w}^{t+1} - \boldsymbol{w}^t\|^2
\end{aligned}$$

Since $\|\boldsymbol{w}^t\|, \|\boldsymbol{w}^{t+1}\| = 1$, we know $(\boldsymbol{w}^{t+1} - \boldsymbol{w}^t)$ bisects the angle between $\boldsymbol{w}^{t+1}$ and $-\boldsymbol{w}^t$. The assumption $\|\eta\mathbb{E}_{\boldsymbol{Z}}[g(\boldsymbol{w}^t, \boldsymbol{Z})] + \eta\beta(\boldsymbol{w}^t - \boldsymbol{u}^{t+1})\| \leq \frac{1}{2}$ guarantees $\frac{2}{3} \leq C^t \leq 2$ and $\theta(-\boldsymbol{w}^t, \boldsymbol{w}^{t+1}) < \pi$. It follows that $\theta(\boldsymbol{w}^{t+1} - \boldsymbol{w}^t, \boldsymbol{w}^t)$ and $\theta(\boldsymbol{w}^{t+1} - \boldsymbol{w}^t, \boldsymbol{w}^{t+1})$ are strictly less than $\frac{\pi}{2}$. On the other hand, $\left(\frac{\boldsymbol{w}^{t+1}}{C^t} - \boldsymbol{w}^t\right)$ also lies in the plane bounded by $\boldsymbol{w}^{t+1}$ and $-\boldsymbol{w}^t$. Therefore

$$\theta\left(\frac{\boldsymbol{w}^{t+1}}{C^t} - \boldsymbol{w}^t, \boldsymbol{w}^{t+1} - \boldsymbol{w}^t\right) < \frac{\pi}{2}.$$

This implies $\langle \frac{\boldsymbol{w}^{t+1}}{C^t} - \boldsymbol{w}^t, \boldsymbol{w}^{t+1} - \boldsymbol{w}^t \rangle \geq 0$. Moreover, when $C^t \geq 1$:

$$\begin{aligned}\langle \frac{\boldsymbol{w}^{t+1}}{C^t} - \boldsymbol{w}^t, \boldsymbol{w}^{t+1} - \boldsymbol{w}^t \rangle =& \langle \frac{\boldsymbol{w}^{t+1}}{C^t} - \frac{\boldsymbol{w}^t}{C^t}, \boldsymbol{w}^{t+1} - \boldsymbol{w}^t \rangle - \langle \frac{C^t - 1}{C^t}\boldsymbol{w}^t, \boldsymbol{w}^{t+1} - \boldsymbol{w}^t \rangle \\ \geq& \frac{1}{C^t}\|\boldsymbol{w}^{t+1} - \boldsymbol{w}^t\|^2\end{aligned}$$

And when $\frac{2}{3} \leq C^t \leq 1$:

$$\begin{aligned}\langle \frac{\boldsymbol{w}^{t+1}}{C^t} - \boldsymbol{w}^t, \boldsymbol{w}^{t+1} - \boldsymbol{w}^t \rangle =& \langle \boldsymbol{w}^{t+1} - \boldsymbol{w}^t, \boldsymbol{w}^{t+1} - \boldsymbol{w}^t \rangle + \langle \frac{1 - C^t}{C^t}\boldsymbol{w}^{t+1}, \boldsymbol{w}^{t+1} - \boldsymbol{w}^t \rangle \\ \geq& \|\boldsymbol{w}^{t+1} - \boldsymbol{w}^t\|^2\end{aligned}$$

Thus we have

$$\mathcal{L}_\beta(\boldsymbol{u}, \boldsymbol{w}^{t+1}) - \mathcal{L}_\beta(\boldsymbol{u}, \boldsymbol{w}^t) \leq \left(\frac{L}{2} + \frac{\beta}{2} - \frac{\chi_{\{C^t \geq 1\}}}{\eta C^t} - \frac{\chi_{\{\frac{2}{3} \leq C^t \leq 1\}}}{\eta}\right)\|\boldsymbol{w}^{t+1} - \boldsymbol{w}^t\|^2$$

Therefore, if η is small so that $\eta \leq \frac{2}{C^t(\beta+L)}$ and $\eta \leq \frac{2}{\beta+L}$, the update on $\boldsymbol{w}$ will decrease $\mathcal{L}_\beta$. Since $C^t \leq 2$, the condition is satisfied when $\eta \leq \frac{1}{\beta+L}$. □

3.4.4 Proof of Lemma 13

Using an argument similar to section 3.3.4, one can show $\theta < \delta$.

Notice that at convergence, after some simplification, we have

$$\left(\boldsymbol{w}^* - \frac{2\sqrt{2\pi}}{k}\beta(\bar{\boldsymbol{w}} - \bar{\boldsymbol{u}})\right) /\!/ \, \bar{\boldsymbol{w}} \tag{3.28}$$

From expression (3.28), we see that $\boldsymbol{w}^*$, after subtracting some vector whose signs agree with $\bar{\boldsymbol{w}}$, and whose non-zero components have the same magnitude $\frac{2\sqrt{2\pi}}{k}\lambda$, is parallel to $\bar{\boldsymbol{w}}$. This implies $\bar{\boldsymbol{w}}$ is some soft-thresholded version of $\boldsymbol{w}^*$, modulo normalization. Moreover, since $\left\|\frac{2\sqrt{2\pi}}{k}\beta(\bar{\boldsymbol{w}} - \bar{\boldsymbol{u}})\right\| \leq \frac{2\sqrt{2\pi}}{k}\lambda\sqrt{d}$, for small λ such that $\frac{2\sqrt{2\pi}}{k}\lambda\sqrt{d} < 1$, we must have

$$\theta\left(\boldsymbol{w}^* - \frac{2\sqrt{2\pi}}{k}\beta(\bar{\boldsymbol{w}} - \bar{\boldsymbol{u}}), \bar{\boldsymbol{w}}\right) = 0$$

On the other hand,

$$\begin{aligned}\left\|\boldsymbol{w}^* - \frac{2\sqrt{2\pi}}{k}\beta(\bar{\boldsymbol{w}} - \bar{\boldsymbol{u}})\right\| &\geq \|\boldsymbol{w}^*\| - \left\|\frac{2\sqrt{2\pi}}{k}\beta(\bar{\boldsymbol{w}} - \bar{\boldsymbol{u}})\right\| \\ &\geq 1 - \frac{2\sqrt{2\pi}}{k}\lambda\sqrt{d}\end{aligned}$$

therefore, $\boldsymbol{w}^* - \frac{2\sqrt{2\pi}}{k}\beta(\bar{\boldsymbol{w}} - \bar{\boldsymbol{u}}) = C\bar{\boldsymbol{w}}$, for some constant C such that $0 < C \leq \frac{k}{k - 2\lambda\sqrt{2\pi d}}$.

Finally, consider the equilateral triangle with sides $\boldsymbol{w}^*, \bar{\boldsymbol{w}}$, and $\boldsymbol{w}^* - \bar{\boldsymbol{w}}$. By the law of sines,

$$\frac{\|\boldsymbol{w}^* - \bar{\boldsymbol{w}}\|}{\sin\theta} = \frac{\|\boldsymbol{w}^*\|}{\sin\theta(\bar{\boldsymbol{w}}, \boldsymbol{w}^* - \bar{\boldsymbol{w}})} = \frac{1}{\sin\theta(\bar{\boldsymbol{w}}, \boldsymbol{w}^* - \bar{\boldsymbol{w}})}$$

as θ is small, $\theta(\bar{\boldsymbol{w}}, \boldsymbol{w}^* - \bar{\boldsymbol{w}})$ is near $\frac{\pi}{2}$. We can assume $\sin\theta(\bar{\boldsymbol{w}}, \boldsymbol{w}^* - \bar{\boldsymbol{w}}) \geq \frac{1}{2}$. Together with the condition $\frac{k}{2\sqrt{2\pi}}\sin\theta = \beta\|\bar{\boldsymbol{u}}\|\sin\gamma$, we have

$$\|\boldsymbol{w}^* - \bar{\boldsymbol{w}}\| \leq 2\sin\theta = \frac{4\sqrt{2\pi}\beta\|\bar{\boldsymbol{u}}\|\sin\gamma}{k} \leq \frac{4\sqrt{2\pi}\beta\sin\gamma}{k}. \qquad \square$$

3.4.5 Proof of Theorem 3.2

Combining Lemmas 8 - 13, Theorem 3.2 is proved. □

3.4.6 Proof of Corollary

Lemma 14. *[76] Let*

$$f_{\lambda,x}(y) = \frac{1}{2}(y-x)^2 + \lambda\, \rho_a(y),$$

$$g_\lambda(x) = sgn(x)\left\{\frac{2}{3}(a+|x|)\cos\left(\frac{\phi(x)}{3}\right) - \frac{2a}{3} + \frac{|x|}{3}\right\}$$

where $\phi(x) = \arccos\left(1 - \frac{27\lambda a(a+1)}{2(a+|x|)^3}\right)$. *Then* $y^*_\lambda(x) = \arg\min_y f_{\lambda,x}(y)$ *is the* $T\ell_1$ *thresholding, equal to* $g_\lambda(x)$ *if* $|x| > t$*; zero elsewhere. Here* $t = \lambda\frac{a+1}{a}$ *if* $\lambda \le \frac{a^2}{2(a+1)}$*;* $t = \sqrt{2\lambda(a+1)} - \frac{a}{2}$*, elsewhere.*

Lemma 15. *[5] Let* $f_{\lambda,x}(y) = \frac{1}{2}(y-x)^2 + \lambda\,\|y\|_0$*. Then* $y^*_\lambda(x) = \arg\min_y f_{\lambda,x}(y)$ *is the* ℓ_0 *hard thresholding* $y^*_\lambda(x) = x$*, if* $|x| > \sqrt{2\lambda}$*; zero elsewhere.*

We proceed by an outline similar to the proof of Theorem 3.2:

Step 1. First we show that $L_{\beta,T\ell_1}(\boldsymbol{u}^t, \boldsymbol{w}^t)$ and $L_{\beta,0}(\boldsymbol{u}^t, \boldsymbol{w}^t)$ both decrease under the update of $\boldsymbol{u}^t$ and $\boldsymbol{w}^t$. To see this, notice that the update on $\boldsymbol{u}^t$ decreases $L_{\beta,T\ell_1}(\boldsymbol{u}^t, \boldsymbol{w}^t)$ and $L_{\beta,0}(\boldsymbol{u}^t, \boldsymbol{w}^t)$ by definition. Then, for a fixed $\boldsymbol{u} = \boldsymbol{u}^{t+1}$, the update on $\boldsymbol{w}^t$ decreases $L_{\beta,T\ell_1}(\boldsymbol{u}^t, \boldsymbol{w}^t)$ and $L_{\beta,0}(\boldsymbol{u}^t, \boldsymbol{w}^t)$ by a similar argument to that found in Theorem 3.2.

Step 2. Next, we show $\theta(\boldsymbol{w}^t, \boldsymbol{w}^*) \le \pi - \delta$, for some $\delta > 0$, for all t, with initialization $\theta(\boldsymbol{w}^0, \boldsymbol{w}^*) = \pi - \delta$. For $L_{\beta,T\ell_1}(\boldsymbol{u}^t, \boldsymbol{w}^t)$, by Lemma 14, we have

$$\boldsymbol{u}^{t+1} = (g_{\lambda/\beta}(w^t_1), g_{\lambda/\beta}(w^t_2), ..., g_{\lambda/\beta}(w^t_d))$$

And for $L_{\beta,0}(\boldsymbol{u}^t, \boldsymbol{w}^t)$, by Lemma 15,

$$\boldsymbol{u}^{t+1} = (w_1^t \chi_{\{|w_1^t| \geq t\}}, w_2^t \chi_{\{|w_2^t| \geq t\}}, ...)$$

In both cases, each component of $\boldsymbol{u}^{t+1}$ is a thresholded version of the corresponding component of $\boldsymbol{w}^t$. This implies $\theta(\boldsymbol{u}^{t+1}, \boldsymbol{w}^t) \leq \frac{\pi}{2}$, and thus the argument in Theorem 3.2 follows through, and we have $\theta(\boldsymbol{w}^t, \boldsymbol{w}^*) \leq \pi - \theta$, for all t.

Step 3. Finally, the equilibrium condition from equation (3.21) still holds for the limit point, and a similar argument shows that $\theta(\bar{\boldsymbol{w}}, \boldsymbol{w}^*) < \delta$. □

3.5 Numerical Experiments

First, we experiment RVSM with VGG-16 on the CIFAR10 data set. Table 3.1 shows the result of RVSM under different penalties. The parameters used are $\lambda = 1.e - 5, \beta = 1.e - 2$, and $a = 1$ for Tℓ_1 penalty. It can be seen that RVSM can maintain very good accuracy while also promotes good sparsity in the trained network. Between the penalties, ℓ_0 gives the best sparsity, ℓ_1 the best accuracy, and Tℓ_1 gives a middle ground between ℓ_0 and ℓ_1. Since the only difference between these parameters is in the pruning threshold, in practice, one may simply stick to ℓ_0 regularization and just fine-tune the hyper-parameters.

Secondly, we experiment our method on ResNet18 and the CIFAR10 data set. The results are displayed in Table 3.2. The base model was trained on 200 epochs using standard SGD method with initial learning rate 0.1, which decays by a factor of 10 at the 80th, 120th, and 160th epochs. For the RVSM method, we use ℓ_0 regularization and set $\lambda =$ 1.e-6, $\beta =$ 8.e-2. For ADMM, we set the pruning threshold to be 60% and $\rho =$1.e-2. The ADMM method implemented here is per [78], an “empirical variation” of the true ADMM (Eq. 3.11). In particular, the arg min update of $\boldsymbol{w}^t$ is replaced by a gradient descent step. Such "modified" ADMM is commonly used in practice on DNN.

It can be seen in Table 3.2 that RVSM runs quite effectively on the benchmark deep network, promote much better sparsity than ADMM (93.70% vs. 47.08%), and has slightly better performance. The sparsity here is the percentage of zero components over all network weights.

Table 3.1: Sparsity and accuracy of RVSM under different penalties on VGG-16 on CIFAR10.

Penalty	Accuracy	Sparsity
Base model	93.82	0
ℓ_1	93.7	35.68
Tℓ_1	93.07	63.34
ℓ_0	92.54	86.89

Table 3.2: Comparison between ADMM and RVSM (ℓ_0) for ResNet18 training on the CIFAR10 dataset.

ResNet18	Accuracy	Sparsity
SGD	95.07	0
ADMM	94.84	47.08
RVSM (ℓ_0)	94.89	93.70

Chapter 4

Generalization of RVSM

4.1 Structured Pruning and Multi-layer Network

4.1.1 The Relaxed Group-wise Splitting Method

In this section, we generalize the framework of RVSM to structured pruning. In the previous chapter, we showed that RVSM can greatly reduce the number of floating point operations (FLOPs) in a network. However, to actually prune and reduce the size of a DNN, it is more convenient if certain structures (for example channels, filters,...) of the network are fully zero. Depending on the hardware limitation, one may choose to prioritize latency over model size, or vice versa.

Consider a one-layer convolution network (not necessarily non-overlap) with weight $\boldsymbol{W}$ and loss function $f(\boldsymbol{W})$. Consider the minimization problem on the Lagrangian:

$$\mathcal{L}_\beta(\boldsymbol{W},\boldsymbol{U}) = f(\boldsymbol{W}) + \lambda P(\boldsymbol{U}) + \frac{\beta}{2}\|\boldsymbol{W} - \boldsymbol{U}\|_2^2 \tag{4.1}$$

where $P(\boldsymbol{U})$ is some regularization function. Suppose $\boldsymbol{W}$ has L channels and let $\boldsymbol{W}_1, \boldsymbol{W}_2, ..., \boldsymbol{W}_L$ (resp. $\boldsymbol{U}_1, ..., \boldsymbol{U}_L$) be the channels of $\boldsymbol{W}$ (resp. $\boldsymbol{U}$). Let $P(\cdot)$ be the Group-Lasso norm [69], equation (4.1) becomes

$$\mathcal{L}_\beta(\boldsymbol{W}, \boldsymbol{U}) = f(\boldsymbol{W}) + \lambda \sum_{i=1}^{L} \|\boldsymbol{U}_i\|_2 + \frac{\beta}{2}(\sum_{i=1}^{L} \|\boldsymbol{W}_i - \boldsymbol{U}_i\|_2^2). \tag{4.2}$$

Let I_g be the indices of $\boldsymbol{W}$ in $\boldsymbol{W}_g$. The solution to

$$\boldsymbol{U}_g^* = \arg\min_{\boldsymbol{U}_g} \left\{ \lambda \|\boldsymbol{U}_g\|_2 + \frac{\beta}{2} \sum_{i \in I_g} \|\boldsymbol{W}_{i,g} - \boldsymbol{U}_{i,g}\|_2^2 \right\} \tag{4.3}$$

is a soft-thresholding operation:

$$U_g^* = \text{Prox}_{GL,\lambda/\beta}(\boldsymbol{W}_g) := \frac{\boldsymbol{W}_g}{\|\boldsymbol{W}_g\|_2} \max\left(\|\boldsymbol{W}_g\|_2 - \frac{\lambda}{\beta}, 0 \right) \tag{4.4}$$

Thus, we can extend RVSM to channel pruning by repeatedly minimizing each group $\boldsymbol{U}_g$, for $g = 1, ..., L$, and applying gradient descent (of the Lagrangian) on $\boldsymbol{W}$. The steps are described in Algorithm 3.

Algorithm 3 RGSM

```
Input: $\eta, \beta, \lambda$, $max_{epoch}$, $max_{batch}$
Initialization: $\boldsymbol{W}^0, \boldsymbol{U}^0$
for $t = 0, 1, 2, ..., max_{epoch}$ do
    for $batch = 1, 2, ..., max_{batch}$ do
        $\boldsymbol{W}^{t+1} \leftarrow \boldsymbol{W}^t - \eta \nabla f(\boldsymbol{W}^t) - \eta\beta(\boldsymbol{W}^t - \boldsymbol{U}^t)$
        for $g = 1, 2, ..., L$ do
            $\boldsymbol{U}_g^{t+1} \leftarrow \arg\min_{\boldsymbol{U}_g} \mathcal{L}_\beta(\boldsymbol{W}^t, \boldsymbol{U}) = \text{Prox}_{GL,\lambda/\beta}(\boldsymbol{W}_g^t)$
        end for
    end for
end for
```

Another useful penalty $P(\cdot)$ is the Group-ℓ_0 norm: $\|\boldsymbol{W}\|_{G\ell_0} = \sum_{g=1}^{L} \mathbf{1}_{\{\|\boldsymbol{W}_g\|_2 \neq 0\}}$. Under the Group-$\ell_0$ norm, the solution to

$$\boldsymbol{U}_g^* = \arg\min_{\boldsymbol{U}_g} \left\{ \lambda \mathbf{1}_{\{\|\boldsymbol{U}_g\|_2 \neq 0\}} + \frac{\beta}{2} \sum_{i \in I_g} \|\boldsymbol{W}_{i,g} - \boldsymbol{U}_{i,g}\|_2^2 \right\} \tag{4.5}$$

is a hard-thresholding operation:

$$\boldsymbol{U}_g^* = \mathrm{Prox}_{G\ell_0, \lambda/\beta}(\boldsymbol{W}_g) := \boldsymbol{W}_g \mathbf{1}_{\{\|\boldsymbol{W}_g\|_2 > \sqrt{2\lambda/\beta}\}} \tag{4.6}$$

and Algorithm 3 can be modified accordingly.

In a similar manner, RVSM/RGSM can be extended to multi-layer DNN by repeatedly applying the algorithm to each layer separately. With some extra assumption on the regularity of the gradient, one can extend the proof in chapter 3 to a general DNN.

4.1.2 Convergence Analysis

We discuss and give a sketch of proof for the convergence result of a multi-layer DNN. The detailed proof is similar to chapter 3.

Assumption 1. *Let $\boldsymbol{W}_1, \boldsymbol{W}_2, ..., \boldsymbol{W}_M$ be the weights in the L layers of a DNN with population loss $f(\boldsymbol{W}_1, \boldsymbol{W}_2, ..., \boldsymbol{W}_M)$. Then there exists a positive constant L such that for all t,*

$$\|\nabla f(..., \boldsymbol{W}_j^{t+1}, ...) - \nabla f(..., \boldsymbol{W}_j^t, ...)\| \leq L \|\boldsymbol{W}_j^{t+1} - \boldsymbol{W}_j^t\| \tag{4.7}$$

for $j = 1, 2, ..., M$.

Assumption 1 is a weaker version of that made by [66, 59], in which the empirical loss function is assumed to be smooth in both the input $\boldsymbol{x}$ and parameters $\boldsymbol{W}$. Here we only require the population loss to be smooth in each layer of the DNN, in the region of iterations. An important consequence of Assumption 1 is:

$$f(..,\boldsymbol{W}_j^{t+1},...)-f(...,\boldsymbol{W}_j^{t},...) \leq \nabla f(...,\boldsymbol{W}_j^{t},...)\cdot(...,\boldsymbol{W}_j^{t+1}-\boldsymbol{W}_j^{t},...)+\frac{L}{2}\|\boldsymbol{W}_j^{t+1}-\boldsymbol{W}_j^{t}\|^2 \quad (4.8)$$

Theorem 4.1. *Suppose Assumption 1 holds, and Algorithm 3 is initiated with step size* $\eta < \frac{2}{\beta+L}$*. Then the Lagrangian* $\mathcal{L}_\beta(\boldsymbol{W}^t,\boldsymbol{U}^t)$ *decreases monotonically and converges subsequentially to a stationary point* $(\bar{\boldsymbol{W}},\bar{\boldsymbol{U}})$*.*

Sketch of Proof: Since RGSM is applied sequentially on each layer, it suffices to consider a one layer network and show $\mathcal{L}_\beta(\boldsymbol{W}^{t+1},\boldsymbol{U}^{t+1}) \leq \mathcal{L}_\beta(\boldsymbol{W}^t,\boldsymbol{U}^t)$. Note that for each group weight, the update of U_g^t decreases the Lagrangian. Therefore $\mathcal{L}_\beta(\boldsymbol{W}^t,\boldsymbol{U}^{t+1}) \leq \mathcal{L}_\beta(\boldsymbol{W}^t,\boldsymbol{U}^t)$. Using a similar argument to section 3 and assumption 1, one can show $\mathcal{L}_\beta(\boldsymbol{W}^{t+1},\boldsymbol{U}^{t+1}) \leq \mathcal{L}_\beta(\boldsymbol{W}^t,\boldsymbol{U}^t)$. This concludes the proof.

4.1.3 Numerical Experiments

First, we implement RGSM on LeNet [35] on the MNIST dataset. The model was trained for 100 epochs using Stochastic GD with momentum 0.9, weight decay 5.e-4, and initial step size 0.1, which is divided by 10 at epoch 60. The results are displayed in Table 4.1. RGSM can achieve higher channel sparsity than [69], with very little performance loss. Notice that a channel sparsity of 71.43% is the highest possible value for LeNet, as there are only 7 channels across the two convolution layers. Therefore, we have shown that RGSM can optimally reduce each convolution layer to one non-zero channel.

Table 4.1: Accuracy and channel sparsity of RGSM on LeNet and MNIST, with parameters $\beta = 1$ and $\lambda = 1$.

Penalty	Accuracy	Channel Sparsity
Base model	99.1	0
RGSM (GL)	98.74	71.43

Next, we extend our experiments on two other standard networks: ResNet18 and VGG16 on the CIFAR10. These models were trained on 200 epochs using Stochastic GD with momentum 0.9, weight decay 5.e-4, and initial step size 0.1, which decays by a factor of 10 at epochs 100 and 160. The results are displayed in Table 4.2. For both models, RGSM can maintain accuracy within 0.5% of the base model, while greatly improving channel sparsity. For ResNet18, almost half (45.8%) of the channels can be pruned off with minimal loss to performance. And since VGG16 is a much larger network, channel sparsity

Table 4.2: Accuracy and channel sparsity of RGSM on ResNet18 and VGG16, on CIFAR10.

Model	Penalty	β	λ	Accuracy	Channel Sparsity
ResNet18	Base model	1	0	94.97	0
	RGSM (GL)	1	1.e-3	94.74	45.8
VGG16	Base model	1	0	93.94	0
	RGSM (GL)	1	1.e-3	93.68	69.0

4.1.4 On The Modern Approach to Pruning

Traditionally, the pruning pipeline involves three major steps: train; prune; and optimize (which may involves retraining). Recently, [41] proposed a new approach: It is better to start with a pruned model and retrain from scratch. Using this method, one can achieve better performance (both accuracy and sparsity) than many modern post-training pruning techniques. In this section, we compare RGSM with this training-from-scratch approach on some standard networks and dataset. Notice that RGSM does not fit into either of these methods; as it trains, prunes, and optimizes the model in one single iteration.

An important question in the implementation of [41] is: Before retraining from scratch, how do we prune a model? If one trains a model until convergence, prunes, and retrains, then this approach is no different than the traditional working pipeline. On the other hand, random initialization pruning requires careful consideration for the network structure. A fixed percentage pruning in each layer is certainly not ideal, as some layers contain more important channels than others.

One notable method for channel selection is Network Slimming [40]. We tested this approach against RGSM for VGG-16 on CIFAR10. With post-training pruning, Network Slimming fails to compile the model at 70% channel sparsity. Their implementation results in layers with 0 channels, thus giving invalid configuration for rebuilding and retraining. With random weight initialization, the results are shown in Table 4.3. RGSM consistently outperforms retrain-from-scratch with Network Slimming at all levels, even at 70% RGSM vs. 50% NS channel sparsity (93.62% vs. 90.91% accuracy).

We also include in Table 4.3 the implementation of ℓ_1-norm pruning [37]. This is a standard train-prune-optimize approach, where the optimization step involves some retraining. It can be seen that RGSM gives better result in both accuracy (93.62% vs. 93.30%) and sparsity (70% vs. 64%).

Finally, as pointed out by [70], each layer may have a different number of unimportant channels. For example, with ResNet18, the last two layers contain so many unimportant channels, over 90% of which could be pruned off; for other layers, this number can be as low as under 10%. With or without retraining, choosing the correct channels to prune is a

Table 4.3: Performance of RGSM against some state-of-the-art pruning algorithms, for VGG-16 on CIFAR10.

VGG-16	Sparsity	Accuracy
Baseline	0	93.96
Network slimming	50%	90.91
Network slimming	60%	90.27
Network slimming	70%	71.88
ℓ_1-norm pruning	64%	93.30
RGSM	**70%**	**93.62**

critical step for model compression. We have shown that RGSM can effectively accomplish this task while also out-performing many state-of-the-art techniques.

4.2 Pruning Robustly Trained Network

In general, networks that are trained against adversarial attacks are less sparse than those naturally trained (Figure 1.2). As a result, traditional post-training pruning methods do not work as well on AT models. In this section, we discuss the common techniques in generating AT models, and show that RVSM/RGSM can still effectively sparsify such networks.

4.2.1 Overview

First, we go over some common adversarial attacks. We focus on the ℓ_∞ norm based untargeted approach. For a given image-label pair $\{\boldsymbol{x}, y\}$ and a network with weights $\boldsymbol{w}$ whose output is $F(\boldsymbol{x}, \boldsymbol{w})$:

- Fast gradient sign method (FGSM) searches an adversarial image $\boldsymbol{x}'$ by maximizing the loss function $\mathcal{L}(\boldsymbol{x}', y) = \mathcal{L}(F(\boldsymbol{x}', \boldsymbol{w}), y))$, subject to the constraint $\|\boldsymbol{x}' - \boldsymbol{x}\|_\infty \leq \epsilon$, where ϵ is the maximum perturbation. Linear approximation method shows that the optimal adversarial image is

$$\boldsymbol{x}' = \boldsymbol{x} + \epsilon \cdot \text{sign}\left(\nabla_{\boldsymbol{x}} L(\boldsymbol{x}, y)\right).$$

- Iterative FGSM (IFGSM^M) [19] iterates FGSM M times with step size α and clip the perturbed image on each step as

  $$\boldsymbol{x}^{(m)} = \text{Clip}_{\boldsymbol{x},\epsilon}\left\{\boldsymbol{x}^{(m-1)} + \alpha \cdot \text{sign}\left(\nabla_{\boldsymbol{x}^{(m-1)}} L(\boldsymbol{x}^{(m-1)}, y)\right)\right\}$$

  where $m = 1, 2, ..., M$, with $\boldsymbol{x}^{(0)} = \boldsymbol{x}$ and $\boldsymbol{x}^{(M)} = \boldsymbol{x}'$.

- C&W attack [10] searches the adversarial image by solving

  $$\min_{\delta} ||\delta||_\infty, \quad \text{subject to} \quad F(\boldsymbol{w}, \boldsymbol{x} + \delta) = t, \ \boldsymbol{x} + \delta \in [0, 1]^d,$$

  where δ is the perturbation and t is the target label.

Recently, [22, 72, 55] showed that there is a relationship between the sparsity of weights in a DNN and its adversarial robustness. Under certain conditions, increasing a model's sparsity can also improve its robustness. For practical implementation, [21] considered a low-rank form of the DNN weight matrix with ℓ_0 constraints on the matrix factors in the adversarial training setting. The training algorithm used is a projected gradient descent (PGD) [45] based on finding the worst adversary. This method, however, only applies to the unstructured (component-wise) setting.

We consider a class of Neural ordinary differential equations (ODE) [11]: the Feynman-Kac formalism principled Robust DNN's [65]. Neural ODE is a DNN structure that uses an ODE to describe the data flow of each input data, rather than having a concrete definition for each layer. Specifically, [65, 64, 39] use the theory of transport equation (TE) to model the flow for the whole input distribution. In particular, from the TE viewpoint, [65] modeled training ResNet [25] as finding the optimal control of the following TE:

$$\begin{cases} \frac{\partial u}{\partial t}(\boldsymbol{x}, t) + G(\boldsymbol{x}, \boldsymbol{w}(t)) \cdot \nabla u(\boldsymbol{x}, t) = 0, \quad \boldsymbol{x} \in \mathbb{R}^d, \\ u(\boldsymbol{x}, 1) = g(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathbb{R}^d, \\ u(\boldsymbol{x}_i, 0) = y_i, \quad \boldsymbol{x}_i \in T, \quad \text{with } T \text{ being the training set.} \end{cases} \tag{4.9}$$

where $G(\boldsymbol{x}, \boldsymbol{w}(t))$ encodes the architecture and weights of the underlying ResNet, $u(\boldsymbol{x}, 0)$ serves as the classifier, $g(\boldsymbol{x})$ is the output activation of ResNet, and y_i is the label of $\boldsymbol{x}_i$.

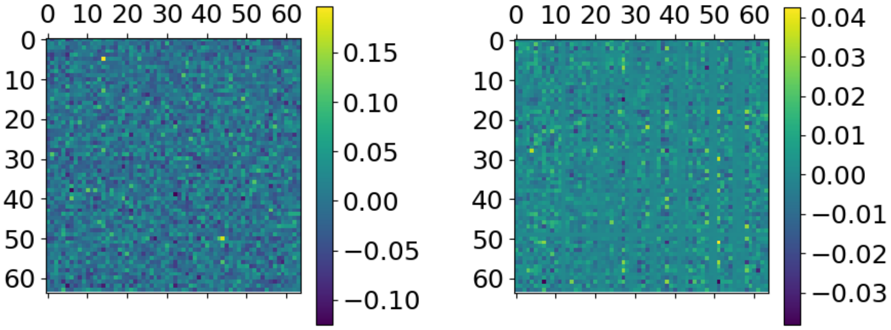
Regarding robustness, [65] interpreted adversarial vulnerability of ResNet as arising from the irregularity of $u(\boldsymbol{x}, 0)$ of the above TE. To enhance $u(\boldsymbol{x}, 0)$'s regularity, one can add a diffusion term, $\frac{1}{2}\sigma^2 \Delta u(\boldsymbol{x}, t)$, to the governing equation of (4.9) which resulting in the convection-diffusion equation (CDE). By the Feynman-Kac formula, $u(\boldsymbol{x}, 0)$ of the CDE can be approximated by the following two steps:

- Modify ResNet by injecting Gaussian noise to each residual mapping;
- Average the output of n jointly trained modified ResNets.

Let En$_n$ResNet denote this ensemble of n ResNets. It was shown in [65] that EnResNet can improve both natural and robust accuracies of the AT networks.

4.2.2 Regularity and Sparsity of the Feynman-Kac Formalism Principled Robust DNNs' Weights

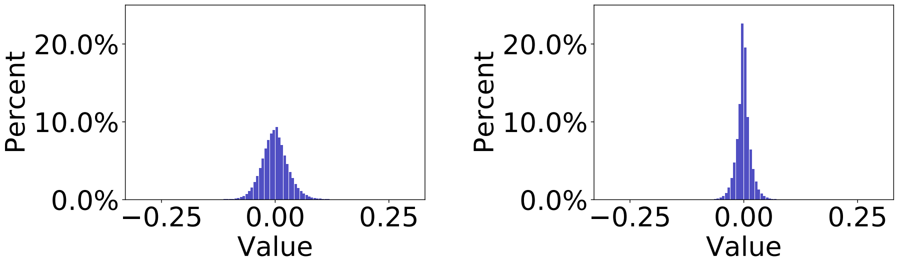
From a partial differential equation (PDE) viewpoint, a diffusion term to the governing equation (4.9) not only smooths $u(\boldsymbol{x}, 0)$, but can also enhance regularity of the velocity field $G(\boldsymbol{x}, \boldsymbol{w}(t))$ [34]. For the DNN counterpart, we expect that when we plot the weights of EnResNet and ResNet at a randomly selected layer, the pattern of the former one will look smoother than the latter. To validate this, we follow the same AT with the same parameters that were used in [65] to train En$_5$ResNet20 and ResNet20, resp. After training, we randomly select and plot the weights of a convolutional layer of ResNet20 whose shape is $64 \times 64 \times 3 \times 3$, and plot the weights at the same layer of the first ResNet20 in En$_5$ResNet20. As shown in Figure 4.1, most of En$_5$ResNet20's weights are close to 0, and they are more regularly distributed in the sense that the neighboring weights are closer to each other than ResNet20's weights. The complete visualization of this randomly selected layer's weights is shown in section A.1.



(a) ResNet20 (AT) (b) En$_5$ResNet20 (AT)

Figure 4.1: Weights visualization

Figure 4.2 shows the weight plot of this convolution layer. The weights of En$_5$ResNet20 are more concentrated around zero than that of ResNet20.



(a) ResNet20 (AT)

(b) En_5ResNet20 (AT)

Figure 4.2: Histogram of weights.

Our approach is to apply the RVSM/RGSM algorithm together with robust PGD training to train and sparsify the model from scratch. Specifically, at each iteration, we apply a PGD attack to generate the adversarial image $\boldsymbol{x}'$, which is then used in the forward-propagation process to generate prediction y'. The back-propagation process will modify loss function to an appropriate Lagrangian and apply RVSM/RGSM accordingly to update the model.

4.2.3 Numerical Results

In this section, we verify that:

- RVSM/RGSM is efficient for unstructured/channel-wise pruning for the AT DNNs, and in general outperforms ADMM-based [78] pruning algorithms.
- After pruning by RVSM and RGSM, EnResNet's weights are significantly more sparse than the baseline ResNet's, and more accurate in classifying both natural and adversarial images.

These two results show that a synergistic integration of RVSM/RGSM with the Feynman-Kac formula principled EnResNet can produce models that meet both sparsity and robustness.

We perform adversarial training by PGD integrated with RVSM/RGSM/ADMM on-the-fly. For all the experiments below, we run 200 epochs of the PGD (10 iterations of the iterative fast gradient sign method (IFGSM^{10}) with $\alpha = 2/255$, $\epsilon = 8/255$, and an initial random perturbation of magnitude ϵ). The initial learning rate is 0.1 and decays by a factor of 10 at the 80th, 120th, and 160th epochs, and the RVSM/RGSM/ADMM sparsification takes place in the back-propagation stage. We split the training data into 45K/5K for training and validation, and the model with the best validation accuracy is used for testing. We test the trained models for both natural accuracy (on clean images) and robustness (against FGSM, IFGSM^{20}, and C&W attacks with the same parameters as that used in [65, 75, 45]). We denote the accuracy on the clean images and under the FGSM, IFGSM^{20} [19], C&W [10], and NAttack [38] attacks as $\boldsymbol{A}_1$, $\boldsymbol{A}_2$, $\boldsymbol{A}_3$, $\boldsymbol{A}_4$, and $\boldsymbol{A}_5$, respectively. We use sparsity for RVSM and channel sparsity for RGSM to measure the performance of the pruning algorithms, where sparsity is defined to be the percentage of zero weights, and channel sparsity is the percentage of channels whose weights' ℓ_2 norm is less than $1.e-15$.

Model Compression for AT ResNet and EnResNets

First, we show that RVSM is efficient to sparsify ResNet and EnResNet. Table 4.4 shows performance of ResNet20 and En$_2$ResNet20 under the unstructured sparsification with $\lambda = 1.e-6$ and varying β. Notice that En$_2$ResNet20 is has better sparsity, accuracy, and robustness than to the baseline ResNet20. For instance, when $\beta = 0.5$, En$_2$ResNet20's weights are 16.42% more sparse than ResNet20's (56.34% vs. 39.92%). Moreover, En$_2$ResNet20 boost the natural and robust accuracies of ResNet20 from 74.08%, 50.64%, 46.67%, and 57.24% to 78.47%, 56.13%, 49.54%, and 65.57%, respectively.

Second, we measure the performance of RGSM in the channel pruning setting. We lists the accuracy and channel sparsity of ResNet20, En$_2$ResNet20, and En$_5$ResNet20 in Table 4.5. Without any sparsification, En$_2$ResNet20 improves the four type of accuracies by 4.27% (76.07% vs. 80.34%), 5.87% (51.24% vs. 57.11%), 2.77% (47.25% vs. 50.02%), and 7.47% (59.30% vs. 66.77%), respectively. For RGSM with $\beta = 1$, $\lambda_1 = 5e-2$, and

Table 4.4: Accuracy and sparsity of ResNet20 and En$_2$ResNet20 under RVSM, with $\lambda = 1.e-6$ and varying β.

	ResNet20					En$_2$ResNet20				
β	$\boldsymbol{A}_1$	$\boldsymbol{A}_2$	$\boldsymbol{A}_3$	$\boldsymbol{A}_4$	Sparsity	$\boldsymbol{A}_1$	$\boldsymbol{A}_2$	$\boldsymbol{A}_3$	$\boldsymbol{A}_4$	Sparsity
n/a	76.07	51.24	47.25	59.30	0	80.34	57.11	50.02	66.77	0
0.01	70.26	46.68	43.79	55.59	80.91	72.81	51.98	46.62	63.10	89.86
0.1	73.45	49.48	45.79	57.72	56.88	77.78	55.48	49.26	65.56	70.55
0.5	74.08	50.64	46.67	57.24	39.92	78.47	56.13	49.54	65.57	56.34

$\lambda_2 = 1.e-5$, both natural and robust accuracies of ResNet20 and En$_2$ResNet20 remain close to the baseline models, but En$_2$ResNet20's weights are 33.48% (41.48% vs. 8%) more sparse than that of ResNet20's. When we increase λ_1 to $1.e-1$, both the accuracy and channel sparsity gaps between ResNet20 and En$_2$ResNet20 are enlarged. En$_5$ResNet20 can future improve both natural and robust accuracies on top of En$_2$ResNet20. For instance, at $\sim 55\%$ (53.36% vs. 56.74%) channel sparsity, En$_5$ResNet20 can improve the four types of accuracy of En$_2$ResNet20 by 4.66% (80.53% vs. 75.87%), 2.73% (57.38% vs. 54.65%), 2.86% (50.63% vs. 47.77%), and 1.11% (66.52% vs. 65.41%), respectively.

Table 4.5: Accuracy and sparsity of different EnResNet20 under RGSM.

Net	β	λ_1	λ_2	$\boldsymbol{A}_1$	$\boldsymbol{A}_2$	$\boldsymbol{A}_3$	$\boldsymbol{A}_4$	$\boldsymbol{A}_5$	Channel Sparsity
ResNet20	n/a	n/a	n/a	76.07	51.24	47.25	59.30	45.88	0
	1	5.e-02	1.e-05	75.91	51.52	47.14	58.77	45.02	8.00
	1	1.e-01	1.e-05	71.84	48.23	45.21	57.09	43.84	25.33
En$_2$ResNet20	n/a	n/a	n/a	80.34	57.11	50.02	66.77	49.35	0
	1	5.e-02	1.e-05	78.28	56.53	49.58	66.56	49.11	41.48
	1	1.e-01	1.e-05	75.87	54.65	47.77	65.41	46.77	56.74
En$_5$ResNet20	n/a	n/a	n/a	81.41	58.21	51.60	66.48	50.21	0
	1	1.e-02	1.e-05	81.46	58.34	51.35	66.84	50.07	19.76
	1	2.e-02	1.e-05	80.53	57.38	50.63	66.52	48.23	53.36

Third, we show that an ensemble of small ResNets via the Feynman-Kac formalism performs better than a larger ResNet of similar size in accuracy, robustness, and sparsity. We apply adversarial training on En$_2$ResNet20 and ResNet38 ($\sim$ 0.54M and $\sim$ 0.56M parameters, respectively), with and without channel pruning. As shown in Table 4.6, under different sets of parameters, after RGSM pruning, En$_2$ResNet20 always has better channel sparsity than ResNet38, and is also more accurate and robust. For instance, when we set $\beta = 1$, $\lambda_1 = 5e{-}2$, and $\lambda_2 = 1.e - 5$, ResNet38 and En$_2$ResNet20 achieve 17.67% and 41.48% channel sparsity, respectively. Moreover, En$_2$ResNet20 outperforms ResNet38 in the four types of accuracy by 0.36% (78.28% vs. 77.92%), 3.02% (56.53% vs. 53.51%), 0.23% (49.58% vs. 49.35%), and 6.34% (66.56% vs. 60.32%), respectively. As shown in Figure 4.3, En$_2$ResNet20's channel sparsity grows much faster than ResNet38's. We verify this observation by plotting the corresponding performances over 5 runs against different λ_1 in Figure 4.4.

Table 4.6: Accuracy and sparsity of En$_2$ResNet20 and ResNet38 under RVSM.

Net	β	λ_1	λ_2	$\boldsymbol{A}_1$	$\boldsymbol{A}_2$	$\boldsymbol{A}_3$	$\boldsymbol{A}_4$	Channel Sparsity
En$_2$ResNet20	n/a	n/a	n/a	**80.34**	**57.11**	**50.02**	**66.77**	0
ResNet38	n/a	n/a	n/a	78.03	54.09	49.81	61.72	0
En$_2$ResNet20	1	5.e-02	1.e-05	**78.28**	**56.53**	**49.58**	**66.56**	**41.48**
ResNet38	1	5.e-02	1.e-05	77.92	53.51	49.35	60.32	17.67
En$_2$ResNet20	1	1.e-01	1.e-05	**76.30**	**54.65**	**47.77**	**65.41**	**56.74**
ResNet38	1	1.e-01	1.e-05	72.95	49.78	46.48	57.92	43.80

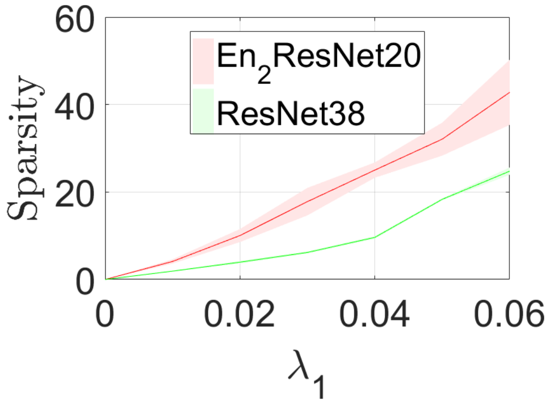


Figure 4.3: Sparsity of En$_2$ResNet20 and ResNet38 under different parameters λ_1.

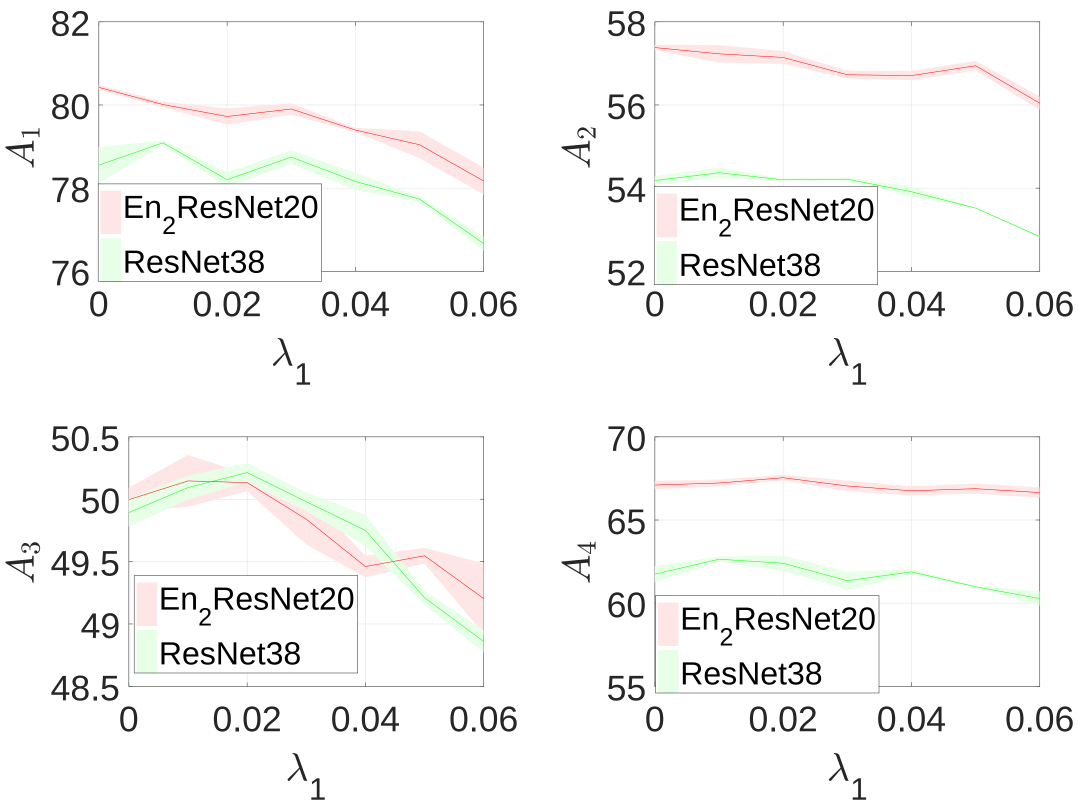


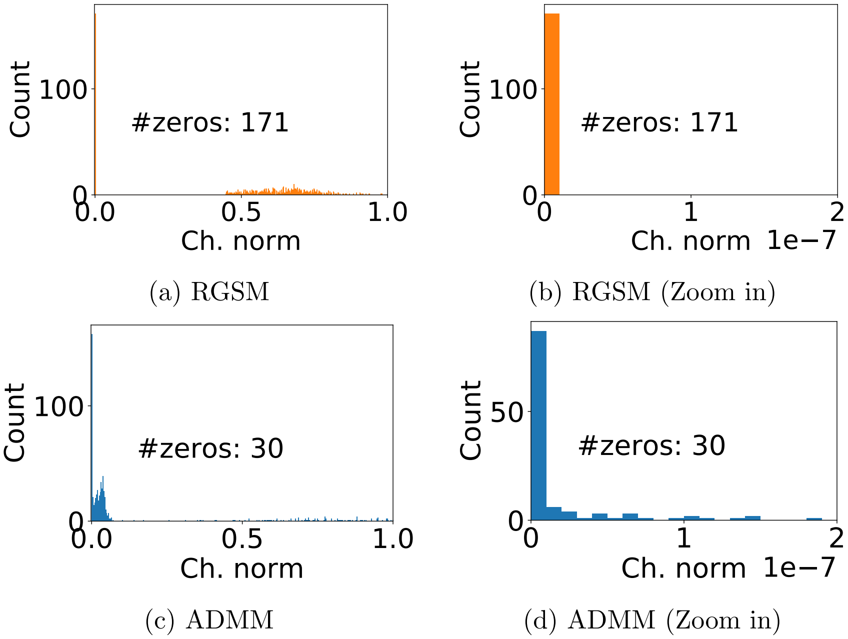
Figure 4.4: Accuracy of En$_2$ResNet20 and ResNet38 under different parameters λ_1.

Table 4.7: Contrasting ADMM versus RVSM (unstructured) and RGSM (channel) for the robustly trained ResNet20.

	Unstructured Pruning					Channel Pruning				
	$\boldsymbol{A}_1$	$\boldsymbol{A}_2$	$\boldsymbol{A}_3$	$\boldsymbol{A}_4$	Sp.	$\boldsymbol{A}_1$	$\boldsymbol{A}_2$	$\boldsymbol{A}_3$	$\boldsymbol{A}_4$	Ch. Sp.
RVSM/RGSM	70.26	46.68	43.79	55.59	**80.91**	**71.84**	**48.23**	**45.21**	**57.09**	**25.33**
ADMM	**71.55**	**47.37**	**44.30**	**55.79**	10.92	63.99	42.06	39.75	51.90	4.44

RVSM/RGSM versus ADMM

In this subsection, we compare the performance of our methods against the classical ADMM [78] for both unstructured and channel pruning settings. For the robustly trained ResNet20, and we will show that RVSM/RGSM can promote much higher sparsity with less natural and robust accuracy degradation than ADMM. We list the performances and sparsities of ResNet20 under ADMM, RVSM, and RGSM in Table 4.7. For unstructured pruning, ADMM retains slightly better natural ($\sim 1.3\%$) and robust ($\sim 0.7\%$, $\sim 0.5\%$, and 0.2% under FGSM, IFGSM^{20}, and C&W attacks) accuracies. However, RVSM gives much better sparsity (80.91% vs. 10.89%). In the channel pruning setting, RGSM significantly outperforms ADMM in all categories, with accuracy improving by at least 5.19% and channel sparsity increasing from 4.44% to 25.33%. Figure 4.5 shows the histograms of channel norms in ResNet20 under RGSM and ADMM. The result agrees with Chapter 3, and verifies our argument from Section 3.1.4. Here, the channel norm is defined to be the ℓ_2 norm of the weights in each channel of the DNN [69].



(a) RGSM

(b) RGSM (Zoom in)

(c) ADMM

(d) ADMM (Zoom in)

Figure 4.5: Channel norms of the AT ResNet20 under RGSM and ADMM.

4.2.4 Beyond CIFAR10

We further show the advantage of applying RVSM/RGSM on EnResNet in compressing and improving accuracy, robustness on the CIFAR100 dataset. The results of ResNet20 and En$_2$ResNet20 are listed in Table 4.8. For $(\beta, \lambda_1, \lambda_2) = (1, 5.e-2, 1.e-5)$, RGSM almost preserves the performance of the baseline model, and also improves channel sparsity by 7.11% for ResNet20, and 16.89% for En$_2$ResNet20. As we increase λ_1 to 0.1, the channel sparsity becomes 18.37% for ResNet20 and 39.23% for En$_2$ResNet20, with less than 3% performance degradation. Without any channel pruning, En$_2$ResNet20 improves natural accuracy by 4.66% (50.68% vs. 46.02%), and robust accuracies by 5.25% (30.2% vs. 24.77%), 3.02% (26.25% vs. 23.23%), and 7.64% (40.06% vs. 32.42%), respectively, under the FGSM, IFGSM^{20}, and C&W attacks. Even in very high channel sparsity scenario ($\lambda_1 = 0.05$), En$_2$ResNet20 still dramatically increase $\boldsymbol{A}_1$, $\boldsymbol{A}_2$, $\boldsymbol{A}_3$, and $\boldsymbol{A}_4$ by 2.90%, 4.31%, 1.89%, and 5.86%, resp. These results are similar to the one obtained on the CIFAR10 in Table 4.5, and further confirm that RGSM together with the Feynman-Kac formalism principled ResNets ensemble can significantly improve both natural and robust accuracy, as well as sparsity of the baseline ResNets.

Table 4.8: Accuracy and sparsity of different Ensembles of ResNet20's on the CIFAR100.

Net	β	λ_1	λ_2	$\boldsymbol{A}_1$	$\boldsymbol{A}_2$	$\boldsymbol{A}_3$	$\boldsymbol{A}_4$	Ch. Sp.
ResNet20	n/a	n/a	n/a	46.02	24.77	23.23	32.42	0
	1	5.e-02	1.e-05	45.74	25.34	23.55	33.53	7.11
	1	1.e-01	1.e-05	44.34	24.46	23.12	32.38	18.37
En$_2$ResNet20	n/a	n/a	n/a	50.68	30.2	26.25	40.06	0
	1	5.e-02	1.e-05	50.56	30.33	26.23	39.85	16.89
	1	1.e-01	1.e-05	47.24	28.77	25.01	38.24	39.23

Chapter 5

Conclusion

We studied the enhanced diffusivity in the perturbed Senile Reinforced Random Walk model. The SeRW model in one dimension with identity reinforcement function was found to be diffusive when perturbed with a small probability δ of breaking out of the last traversed edge, no matter how small δ is. The enhanced diffusivity is logarithmically close to residual diffusivity as δ tends to zero. We studied a few variations of the perturbed models, where the perturbation $\delta\ \xi_n$ is stochastic, and the distribution of ξ_n may or may not depend on n. These models intend to create a "fat tail" as n increases so it is more likely for the walk to break out of the last traversed edge. For most cases, the enhanced diffusivity is $\nu_\delta = O\left(\frac{1}{|\log \delta|}\right)$. The highest enhanced diffusivity is $\nu_\delta = O\left(\frac{1}{\log|\log \delta|}\right)$. This was achieved when ξ_n has a very fat tail, $f_{\xi_n}(x) = O\left(\frac{1}{x(\log x)^2}\right)$, which is much fatter than that of the Cauchy distribution. In higher dimensions, the baseline SeRW with identity reinforcement function is already diffusive and the enhanced diffusivity reaches a rate as high as $O(\log^{-2}\delta)$.

Next, we studied the problem of complexity reduction for deep neural networks (DNN's) via regularization. We propose a Relaxed Variables Splitting Method (RVSM) to regularize DNN's and improve sparsity in the network weights. We proved the global convergence of

RVSM for a one-layer convolution network on a regression problem, analyzed the sparsity of the limiting weight vector and its error estimate from the ground truth (i.e. the global minimum), and demonstrated the effectiveness on training multi-layer models via numerical experiments. The proof used geometric argument to establish angle and Lagrangian descent properties of the iterations thereby overcame the non-existence of gradient at the origin of the population loss function. Our experimental results provided additional support for the effectiveness of RVSM via ℓ_0, ℓ_1 and Tℓ_1 penalties on standard DNN's on the CIFAR10 dataset.

Finally, we generalized the RVSM algorithm to structured pruning and studied its application to adversarial training of DNN's. With structured sparsity, the non-essential weight groups can be safely pruned off without any performance degradation, resulting in a model with smaller size and faster inference rate. We showed through experiments that our generalization holds, one can prune off 70% of the channels in VGG16 with minimal accuracy loss on the CIFAR10 dataset. For robustly trained network, we discussed some common adversarial attacks and how our method can be incorporated into the adversarial training process. For numerical experiments, we tested our approach on the Feynman-Kac formalism principled ResNet ensembles. The result verified that one can create a model that is both sparse and robust via RVSM/RGSM: Compared to traditional ResNets that are robustly trained, our method results in a model with half the size and better performance, in both natural accuracy and robustness.

Our proposed method of DNN's regularization combines the standard three-step pruning pipeline (train, prune, fine-tune) into one, and provides a competitive alternative to other state-of-the-art pruning techniques. One can apply quantization after regularization to further compress the network. For example, on the CIFAR10, the VGG16 can be reduced over 3x in size using RGSM; with 8-bit quantization, the model can be further compressed by 4x, resulting in a model that is over 12x smaller, and has much faster inference rate.

Bibliography

[1] M. Abramowitz and I. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables.* Martino Publishing, 2014.

[2] A. Athalye, N. Carlini, and D. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 274–283, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[3] A. Bensoussan, J.-L. Lions, and G. Papanicolaou. *Asymptotic analysis for periodic structures.* AMS Chelsea Publishing, 2011.

[4] L. Biferale, A. Cristini, M. Vergassola, and A. Vulpiani. Eddy diffusivities in scalar transport. *Physics Fluids*, 7(11):2725–2734, 1995.

[5] T. Blumensath. Accelerated iterative hard thresholding. *Signal Processing*, 92(3):752–756, 2012.

[6] T. Blumensath and M. Davies. Iterative thresholding for sparse approximations. *Journal Of Fourier Analysis And Applications*, 14(5-6):629–654, 2008.

[7] A. Brutzkus and A. Globerson. Globally optimal gradient descent for a convnet with gaussian inputs. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 605–614, Sydney, NSW, Australia, 2017. JMLR.org.

[8] R. Camassa and S. Wiggins. Chaotic advection in a rayleigh-bénard flow. *Physical Review A*, 43(2):774–797, 1990.

[9] E. Candès, J. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 59(8):1207–1223, 2006.

[10] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. *IEEE European Symposium on Security and Privacy*, pages 39–57, 2016.

[11] T. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud. Neural ordinary differential equations. In *Advances in neural information processing systems*, pages 6571–6583, 2018.

[12] Y. Cheng, D. Wang, P. Zhou, and T. Zhang. A survey of model compression and acceleration for deep neural networks. *CoRR*, abs/1710.09282, 2017.

[13] Y. Cho and L. K. Saul. Kernel methods for deep learning. *In Advances in neural information processing systems*, pages 342–350, 2009.

[14] I. Daubechies, M. Defrise, and C. D. Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57(11):1413–1457, 2004.

[15] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier. Language modeling with gated convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 933–941, Sydney, NSW, Australia, 2017. JMLR.org.

[16] D. Donoho. Denoising by soft-thresholding,. *IEEE Transactions on Information Theory*, 41(3):613–627, 1995.

[17] A. Fannjiang and G. Papanicolaou. Convection enhanced diffusion for periodic flows. *SIAM Journal on Applied Mathematics*, 54(2):333–408, 1994.

[18] A. Fannjiang and G. Papanicolaou. Diffusion in turbulence. *Probability Theory and Related Fields*, 105:279–334, 1996.

[19] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In Y. Bengio and Y. LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[20] Google. Gemmlowp: a small self-contained low-precision gemm library. `https://github.com/google/gemmlowp`.

[21] S. Gui, H. Wang, H. Yang, C. Yu, Z. Wang, and J. Liu. Model compression with adversarial robustness: A unified optimization framework. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, Vancouver, BC, Canada*, pages 1283–1294, 2019.

[22] Y. Guo, C. Zhang, C. Zhang, and Y. Chen. Sparse dnns with improved adversarial robustness. In *Advances in neural information processing systems*, pages 242–251, 2018.

[23] S. Han, H. Mao, and W. J. Dally. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. In Y. Bengio and Y. LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico*, 2016.

[24] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

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

[26] S. Heinze. Diffusion-advection in cellular flows with large peclet numbers. *Archive for Rational Mechanics and Analysis volume*, 168(4):329–342, 2003.

[27] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

[28] M. Holmes and A. Sakai. Senile reinforced random walks. *Stochastic Processes and Their Applications, Science Direct*, 20, February 2007.

[29] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017.

[30] H. Kesten and G. Papanicolaou. A limit theorem for turbulent diffusion. *Communications in Mathematical Physics*, 65:97–128, 1979.

[31] R. Kraichnan. Diffusion in a random velocity field. *The Physics of Fluids*, 13:22–31, 1970.

[32] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In P. L. Bartlett, F. C. N. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pages 1106–1114, 2012.

[33] N. Kumar, U. Harbola, and K. Lindenberg. Memory-induced anomalous dynamics: Emergence of diffusion, subdiffusion, and superdiffusion from a single random walk model. *Physical Review E*, 82:021101, 2010.

[34] O. Ladyženskaja, V. Solonnikov, and N. Ural'ceva. *Linear and quasi-linear equations of parabolic type*, volume 23. American Mathematical Soc., 1988.

[35] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998.

[36] Y. LeCun, J. Denker, and S. Solla. Optimal brain damage. In *Proceedings of the 2nd International Conference on Neural Information Processing Systems*, NIPS'89, page 598–605, Cambridge, MA, USA, 1989. MIT Press.

[37] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf. Pruning filters for efficient convnets. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[38] Y. Li, L. Li, L. Wang, T. Zhang, and B. Gong. NATTACK: learning the distributions of adversarial examples for an improved black-box attack on deep neural networks. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 3866–3876. PMLR, 2019.

[39] Z. Li and Z. Shi. Deep residual learning and pdes on manifold. *arXiv preprint arXiv:1708.05115*, 2017.

[40] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, and C. Zhang. Learning efficient convolutional networks through network slimming. In *ICCV*, 2017.

[41] Z. Liu, M. Sun, T. Zhou, G. Huang, and T. Darrell. Rethinking the value of network pruning. In *ICLR*, 2019.

[42] C. Louizos, M. Welling, and D. Kingma. Learning sparse neural networks through l_0 regularization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada*, 2018.

[43] J. Lyu, J. Xin, and Y. Yu. Computing residual diffusivity by adaptive basis learning via spectral method. *Numerical Mathematics: Theory, Methods & Applications*, 10(2):351–372, 2017.

[44] J. Lyu, J. Xin, and Y. yu. Residual diffusivity in elephant random walk models with stops. *Communications in Mathematical Sciences*, 16, 05 2017.

[45] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

[46] A. Majda and P. Kramer. Simplified models for turbulent diffusion: Theory, numerical modelling, and physical phenomena. *Physics Reports*, 314:237–574, 1999.

[47] I. MKL-DNN. Intel(r) math kernel library for deep neural networks. `https://intel.github.io/mkl-dnn/index.html`.

[48] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani. Deep learning for iot big data and streaming analytics: A survey. *IEEE Communications Surveys & Tutorials*, 20(4):2923–2960, 2018.

[49] D. Molchanov, A. Ashukha, and D. Vetrov. Variational dropout sparsifies deep neural networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 2498–2507, Sydney, NSW, Australia, 2017. JMLR.org.

[50] J.-J. Moreau. Proximité et dualité dans un espace hilbertien. *Bulletin de la Société Mathématique de France*, 93:273–299, 1965.

[51] N. Murphy, E. Cherkaev, J. Zhu, J. Xin, and K. Golden. Spectral analysis and computation of effective diffusivities in space-time periodic incompressible flows. *Annals of Mathematical Sciences and Applications*, 2(1):3–66, 2017.

[52] M. Nikolova. Local strong homogeneity of a regularized estimator. *SIAM Journal on Applied Mathematics*, 61(2):633–658, 2000.

[53] A. Novikov, G. Papanicolaou, and L. Ryzhik. Boundary layers for cellular flows at high péclet numbers. *Communications on Pure and Applied Mathematics*, 58(7):867–922, 2005.

[54] NVIDIA. 8 bit inference with tensorrt. `http://on-demand.gputechconf.com/gtc/2017/presentation/s7310-8-bit-inference-with-tensorrt.pdf`.

[55] A. Rakin, Z. He, L. Yang, Y., L. Wang, and D. Fan. Robust sparse regularization: Simultaneously optimizing neural network robustness and compactness. *arXiv preprint arXiv:1905.13074*, 2019.

[56] G. M. Schütz and S. Trimper. Elephants can always remember: Exact long-range memory effects in a non-markovian random walk. *Physical Review E*, 70:045101, Oct 2004.

[57] O. Shamir. Distribution-specific hardness of learning neural networks. *Journal of Machine Learning Research*, 19(1):1135–1163, Jan. 2018.

[58] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[59] A. Sinha, H. Namkoong, and J. Duchi. Certifiable distributional robustness with principled adversarial training. In *International Conference on Learning Representations*, 2018.

[60] G. I. Taylor. Diffusion by continuous movements. *Proceedings of the London Mathematical Society*, s2-20(1):196–212, 1922.

[61] Y. Tian. An analytical formula of population gradient for two-layered relu network and its applications in convergence and critical point analysis. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 3404–3413, Sydney, NSW, Australia, 2017. JMLR.org.

[62] A. M. TURING. I.—COMPUTING MACHINERY AND INTELLIGENCE. *Mind*, LIX(236):433–460, 10 1950.

[63] K. Ullrich, E. Meeds, and M. Welling. Soft weight-sharing for neural network compression. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France*, 2017.

[64] B. Wang, X. Luo, Z. Li, W. Zhu, Z. Shi, and S. Osher. Deep neural nets with interpolating function as output activation. In *Advances in Neural Information Processing Systems*, pages 743–753, 2018.

[65] B. Wang, B. Yuan, Z. Shi, and S. Osher. ResNet ensemble via the Feynman-Kac formalism to improve natural and robust acurcies. In *Advances in Neural Information Processing Systems*, 2019.

[66] Y. Wang, X. Ma, J. Bailey, J. Yi, B. Zhou, and Q. Gu. On the convergence and robustness of adversarial training. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6586–6595, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[67] Y. Wang, J. Zeng, and W. Yin. Global convergence of admm in nonconvex nonsmooth optimization. *J. Sci. Comput.*, 78(1):29–63, Jan. 2019.

[68] Z. Wang, J. Xin, and Z. Zhang. Computing effective diffusivity of chaotic and stochastic flows using structure-preserving schemes. *SIAM Journal on Numerical Analysis*, 56(4):2322–2344, 2018.

[69] W. Wen, C. Wu, Y. Wang, Y. Chen, and H. Li. Learning structured sparsity in deep neural networks. In *Advances in neural information processing systems*, pages 2074–2082, 2016.

[70] B. Yang, J. Lyu, S. Zhang, Y.-Y. Qi, and J. Xin. Channel pruning for deep neural networks via a relaxed group-wise splitting method. *In Proc. of 2nd International Conference on AI for Industries (AI4I), Laguna Hills, CA*, 2019.

[71] S. Yao, Y. Zhao, A. Zhang, L. Su, and T. Abdelzaher. Deepiot: Compressing deep neural network structures for sensing systems with a compressor-critic framework. In *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*, page 4. ACM, 2017.

[72] S. Ye, K. Xu, S. Liu, H. Cheng, J. Lambrechts, H. Zhang, A. Zhou, K. Ma, Y. Wang, and X. Lin. Second rethinking of network pruning in the adversarial setting. *arXiv preprint arXiv:1903.12561*, 2019.

[73] P. Yin, S. Zhang, J. Lyu, S. Osher, Y. Qi, and J. Xin. Blended coarse gradient descent for full quantization of deep neural networks. *Research in the Mathematical Sciences*, 6(14), 2019.

[74] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France*, 2017.

[75] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. E. Ghaoui, and M. Jordan. Theoretically principled trade-off between robustness and accuracy. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 7472–7482, Long Beach, California, USA, 2019. PMLR.

[76] S. Zhang and J. Xin. Minimization of transformed l_1 penalty: Closed form representation and iterative thresholding algorithms. *Communications in Mathematical Sciences*, 15(2):511–537, 2017.

[77] S. Zhang and J. Xin. Minimization of transformed l_1 penalty: Theory, difference of convex function algorithm, and robust application in compressed sensing. *Mathematical Programming, Series B*, 169(1):307–336, 2018.

[78] T. Zhang, S. Ye, K. Zhang, J. Tang, W. Wen, M. Fardad, and Y. Wang. A systematic DNN weight pruning framework using alternating direction method of multipliers. In V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, editors, *Computer Vision – ECCV 2018*, pages 191–207, Cham, 2018. Springer International Publishing.

[79] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le. Learning transferable architectures for scalable image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[80] P. Zu, L. Chen, and J. Xin. A computational study of residual KPP front speeds in time-periodic cellular flows in the small diffusion limit. *Physica D: Nonlinear Phenomena*, 311:37–44, 2015.

Appendix A

Appendix Title

A.1 More Visualizations of the DNNs' Weights

In section 4.2.2, we showed some visualization results for a portion of the weights of a randomly selected convolutional layer of the robustly trained ResNet20 and En_5ResNet20. The complete visualization of this layer is shown in Figs. A.1 and A.2. It can be seen that En_5ResNet20's weights generally have smaller magnitude and more regular weight distribution than that of ResNet20.

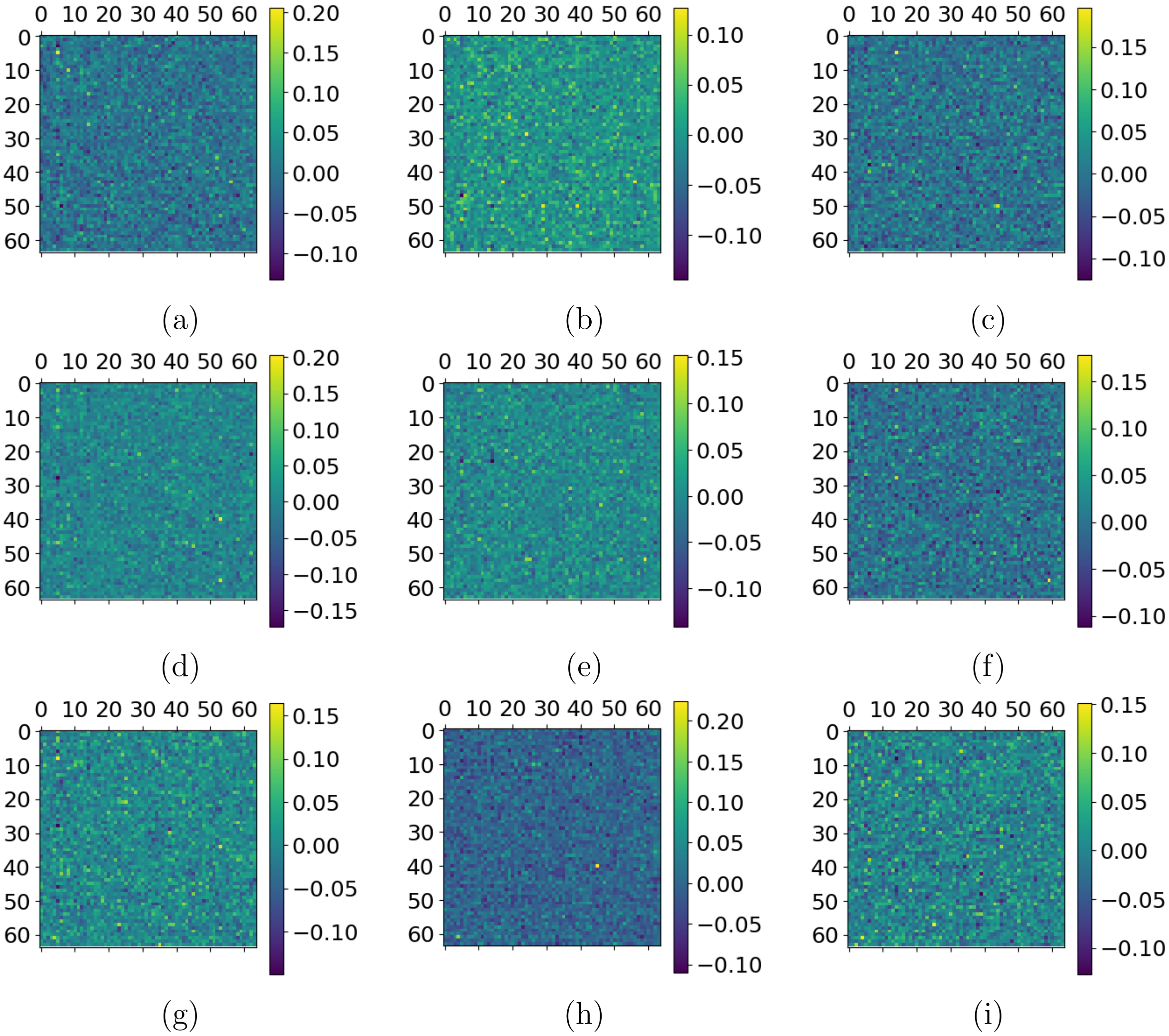
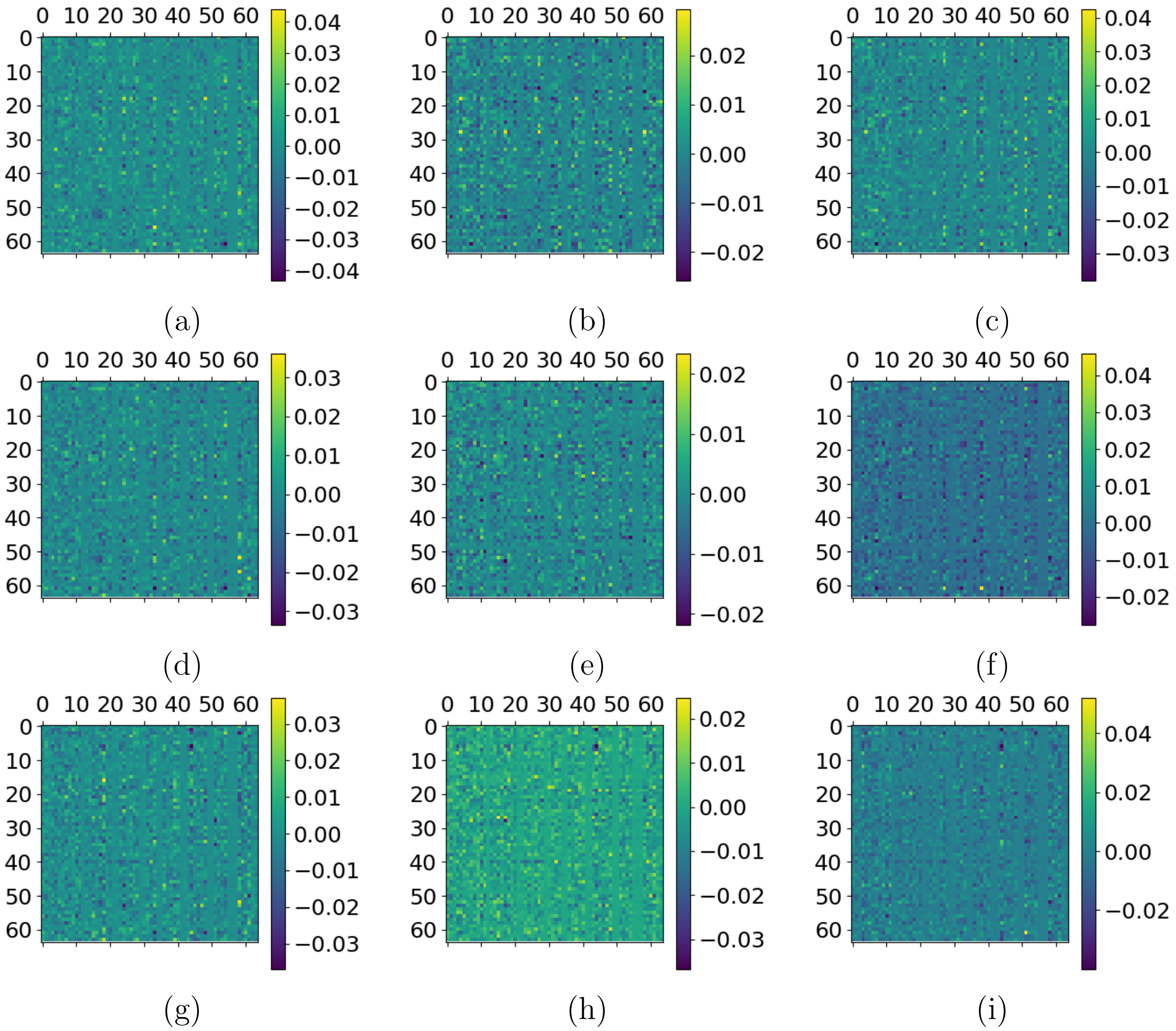


Figure A.1: Weights of a randomly selected convolutional layer of the PGD AT ResNet20.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure A.2: Weights of the PGD AT En_5ResNet20 at the same layer as that shown in Figure A.1.